

**AD-A275 375**Public Re-  
quest for Infor-  
mation  
Commission  
Davis/McC

IN PAGE

Form Approved  
OMB No 0704-0188

Please do not respond including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggested modifications or new collection techniques, to the Information Resources Services Directorate for Information Operations and Reports, 1215 Jefferson Avenue and Building Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AG:	9 DEC 93	3. REPORT TYPE AND DATES COVERED THESIS/DISSERTATION
4. TITLE AND SUBTITLE EXPERIMENTAL EVALUATION OF FUZZY LOGIC CONTROL OF A FLEXIBLE ARM MANIPULATOR		5. FUNDING NUMBERS
6. AUTHOR(S) JOEL JACKSON		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AFIT Student Attending: UNIVERSITY OF WASHINGTON		8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/CI/CIA- 93-167
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) DEPARTMENT OF THE AIR FORCE AFIT/CI 2950 P STREET WRIGHT-PATTERSON AFB OH 45433-7765		10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES		
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for Public Release IAW 190-1 Distribution Unlimited MICHAEL M. BRICKER, SMSgt, USAF Chief Administration		12b. DISTRIBUTION CODE
13. ABSTRACT (Maximum 200 words)		
94-03994 		
94 2 03 208		
14. SUBJECT TERMS		15. NUMBER OF PAGES 76
		16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT
		20. LIMITATION OF ABSTRACT

93-167

Experimental Evaluation of  
Fuzzy Logic Control  
of a  
Flexible Arm Manipulator

by

Joel Jackson

A thesis submitted in partial fulfillment  
of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

University of Washington

1993

Approved by

  
(Chairperson of Supervisory Committee)

Program Authorized

to Offer Degree Department of Aeronautics & Astronautics

Date

9 DEC 93

University of Washington

Abstract

Experimental Evaluation of Fuzzy Logic Control  
of a Flexible Arm Manipulator

by Joel D. Jackson

Chairperson of Supervisory Committee: Professor Juris Vagners

Department of Aeronautics and Astronautics

The purpose of the research reported in this thesis was to explore fuzzy logic control of a single link, flexible robot arm. The performance and stability robustness properties of fuzzy logic control were evaluated in simulation and with hardware in the loop in the University of Washington Control Systems Laboratory. The plant model and the Integrated System, Inc., AC100-C30 real time control prototyping system were first validated using a PID controller, which was also used as a comparison for the performance of various fuzzy logic controllers. The control objective was to slew the arm rapidly to commanded positions while minimizing the excitation of the link's flexible modes. The available sensor was a hub angle sensor (no tip position or tip rate information was used for the controllers). The stability robustness of the PID controller and the fuzzy logic controller were evaluated in hardware for mismodeled system parameters. Tests were conducted for increased tip mass relative to the design condition controller as well as for a significantly more flexible link than the design condition. The fuzzy controller was found to be stable for all test conditions, while the PID controller was unstable for one of the off-design conditions.

DTIC QUALITY INSPECTED 5

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Email and/or Special
A-1	

University of Washington

Abstract

Experimental Evaluation of Fuzzy Logic Control  
of a Flexible Arm Manipulator

by Joel D. Jackson

Chairperson of Supervisory Committee: Professor Juris Vagners

Department of Aeronautics and Astronautics

The purpose of the research reported in this thesis was to explore fuzzy logic control of a single link, flexible robot arm. The performance and stability robustness properties of fuzzy logic control were evaluated in simulation and with hardware in the loop in the University of Washington Control Systems Laboratory. The plant model and the Integrated System, Inc., AC100-C30 real time control prototyping system were first validated using a PID controller, which was also used as a comparison for the performance of various fuzzy logic controllers. The control objective was to slew the arm rapidly to commanded positions while minimizing the excitation of the link's flexible modes. The available sensor was a hub angle sensor (no tip position or tip rate information was used for the controllers). The stability robustness of the PID controller and the fuzzy logic controller were evaluated in hardware for mismodeled system parameters. Tests were conducted for increased tip mass relative to the design condition controller as well as for a significantly more flexible link than the design condition. The fuzzy controller was found to be stable for all test conditions, while the PID controller was unstable for one of the off-design conditions.

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive coping of this thesis is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Any other reproduction for any purposes or by any means shall not be allowed without my written permission.

Signature\_\_\_\_\_

Date\_\_\_\_\_

University of Washington

**Abstract**

**Experimental Evaluation of Fuzzy Logic Control  
of a Flexible Arm Manipulator**

by Joel D. Jackson

Chairperson of Supervisory Committee: Professor Juris Vagners

Department of Aeronautics and Astronautics

The purpose of the research reported in this thesis was to explore fuzzy logic control of a single link, flexible robot arm. The performance and stability robustness properties of fuzzy logic control were evaluated in simulation and with hardware in the loop in the University of Washington Control Systems Laboratory. The plant model and the Integrated System, Inc., AC100-C30 real time control prototyping system were first validated using a PID controller, which was also used as a comparison for the performance of various fuzzy logic controllers. The control objective was to slew the arm rapidly to commanded positions while minimizing the excitation of the link's flexible modes. The available sensor was a hub angle sensor (no tip position or tip rate information was used for the controllers). The stability robustness of the PID controller and the fuzzy logic controller were evaluated in hardware for mismodeled system parameters. Tests were conducted for increased tip mass relative to the design condition controller as well as for a significantly more flexible link than the design condition. The fuzzy controller was found to be stable for all test conditions, while the PID controller was unstable for one of the off-design conditions.

## TABLE OF CONTENTS

	Page
List of Figures .....	ii
List of Tables.....	iv
Chapter 1: Introduction.....	1
Chapter 2: Model Structure.....	6
Chapter 3: Tests and Results .....	12
Chapter 4: Conclusions and Recommendations.....	44
Bibliography .....	47
Appendix A: Plant Generation Programs.....	48
Appendix B: TLA Model Development.....	54
Appendix C: Model Poles and Zeros.....	58
Appendix D: Bode/Root Locus Plots.....	71

## LIST OF FIGURES

Number	Page
1.1 Experimental Configuration.....	3
2.1 System State Description.....	8
2.2 Finite Element Representation.....	9
2.3 Open Loop Poles of the 40 State System.....	10
2.4 Open Loop Poles of Reduced Model.....	11
3.1 Feedback Configuration.....	12
3.2 Smooth Ramp Reference Theta to 15 Degrees, $T_{slew}=1$ sec.....	13
3.3 PID Simulation Theta 10 Degree Slew.....	14
3.4 PID Simulation Torque 10 Degree Slew.....	15
3.5 PID Simulation Theta 15 Degree Slew.....	15
3.6 PID Simulation Torque 15 Degree Slew.....	16
3.7 PID Hardware Theta 10 Degree Slew.....	17
3.8 PID Hardware Torque 10 Degree Slew.....	17
3.9 PID Hardware Theta 15 Degree Slew.....	18
3.10 PID Hardware Torque 15 Degree Slew.....	18
3.11 Membership Curve Concept.....	20
3.12 Multiple Membership Curves.....	21
3.13 Implication.....	22
3.14 Fuzzy Feedback .....	23
3.15 Membership Curve Shapes for Angle Error Velocity and Torque.....	24
3.16 Membership Curve Shapes for Angle Error.....	25
3.17 Five by Five Theta Simulation, 15 Degree Slew.....	26
3.18 Five by Five Torque Simulation, 15 Degree Slew.....	27
3.19 Five by Five Theta in Hardware, 15 degree Slew.....	28
3.20 Five by Five Torque in Hardware, 15 Degree Slew.....	28
3.21 Theta Response for Double Control Law in Hardware, 15 Degree Slew.....	31
3.22 Torque Response for Double Control Law in Hardware, 15 Degree Slew.....	31
3.23 Theta Response for Fuzzy Logic Robustness Test I, 15 Degree Slew.....	32
3.24 Torque Response for Fuzzy Logic Robustness Test I, 15 Degree Slew.....	33
3.25 Theta Response for PID Robustness Test I, 15 Degree Slew.....	33
3.26 Torque Response for PID Robustness Test I, 15 Degree Slew.....	34

3.27 Theta Response for Fuzzy Logic Robustness Test II, 15 Degree Slew.....	35
3.28 Torque Response for Fuzzy Logic Robustness Test II, 15 Degree Slew.....	35
3.29 Theta Response for PID Robustness Test II, 15 Degree Slew.....	36
3.30 Torque Response for PID Robustness Test II, 15 Degree Slew.....	36
3.31 Theta Response for Fuzzy Logic Robustness Test III, 15 Degree Slew.....	37
3.32 Torque Response for Fuzzy Logic Robustness Test III, 15 Degree Slew.....	38
3.33 Theta Response for PID Robustness Test III, 15 Degree Slew.....	38
3.34 Torque Response for PID Robustness Test III, 15 Degree Slew.....	39
3.35 Seven by Seven Theta in Simulation, 15 Degree Slew.....	41
3.36 Seven by Seven Torque in Simulation, 15 Degree Slew.....	41
3.37 Double Seven by Seven Theta in Simulation, 15 Degree Slew.....	43
3.38 Double Seven by Seven Torque in Simulation, 15 Degree Slew.....	43
D.1 Open Loop On-Design Bode Plot.....	71
D.2 Open Loop Off-Design Flexible Link Bode Plot.....	72
D.3 Open Loop On-Design Root Locus.....	73
D.4 Closed Loop PID On-Design Bode Plot.....	74
D.5 Closed Loop PID Off-Design Flexible Link Bode Plot.....	75
D.6 Closed Loop PID On-Design Root Locus.....	76

## LIST OF TABLES

Number	Page
3.1 PID Gains.....	14
3.2 Conjunction of Minimum.....	21
3.3 Angle Error Velocity and Torque Five by Five Membership Curve Equations	24
3.4 Angle Error Five by Five Membership Curve Equations.....	25
3.5 Single Five by Five Controller Rule Matrix .....	25
3.6 Angle Error Large Five by Five Membership Curve Equations.....	30
3.7 Large Five by Five Controller Rule Matrix.....	30
3.8 Single Seven by Seven Controller Membership Curves.....	40
3.9 Single Seven by Seven Controller Rule Matrix.....	40
3.10 Small Seven by Seven Controller Rule Matrix.....	42
4.1 Performance Response.....	45

#### **ACKNOWLEDGMENTS**

I would like to thank Professor Juris Vagners for helping me grasp the big picture and directing me in the way I should go. Also, I want to thank Steve Evers and Dave Bossert. Without their help this thesis would never have gotten off the ground.

## Chapter 1: Introduction

### 1.1 Background

Robots have become common place in today's industrial environment. Their ability to perform repetitive tasks with precision has proved their place in industry. As with any relatively recent technology, however, there continue to be problems as well as advances. Although robots have shown that they are useful as well as cost effective in many industries, the size, weight, and cost of current robots has excluded them from many tasks. To increase the performance, as well as reduce the price, smaller and lighter robots are being developed and their performance evaluated. This reduction in size and weight introduces the problem of structural rigidity for the robot arm or arm segments.

When controlling a robot arm there are two major performance criteria: accuracy and speed. Using a rigid arm, it is generally an easy task to slew a manipulator from point A to point B within a reasonable time. When a lighter, more flexible arm is used, however, the dynamics of the arm may be excited when the same slew time is used. With this added complication, a flexible robot is often unable to complete a given task as quickly as a more sturdy manipulator.

Another problem for the flexible arm is a changing, or unknown plant or payload. Although this problem is not isolated to flexible manipulators, the more excitable dynamics often magnify the change from the design point. To give the robustness that this situation requires, fuzzy logic is employed for the development of the controller logic. Fuzzy logic based controllers are able to offer this robustness with the convenience of intuitive thought and without the large memory requirement of lookup tables or the large computation requirement of formula based solutions [2].

### 1.2 Problem Definition

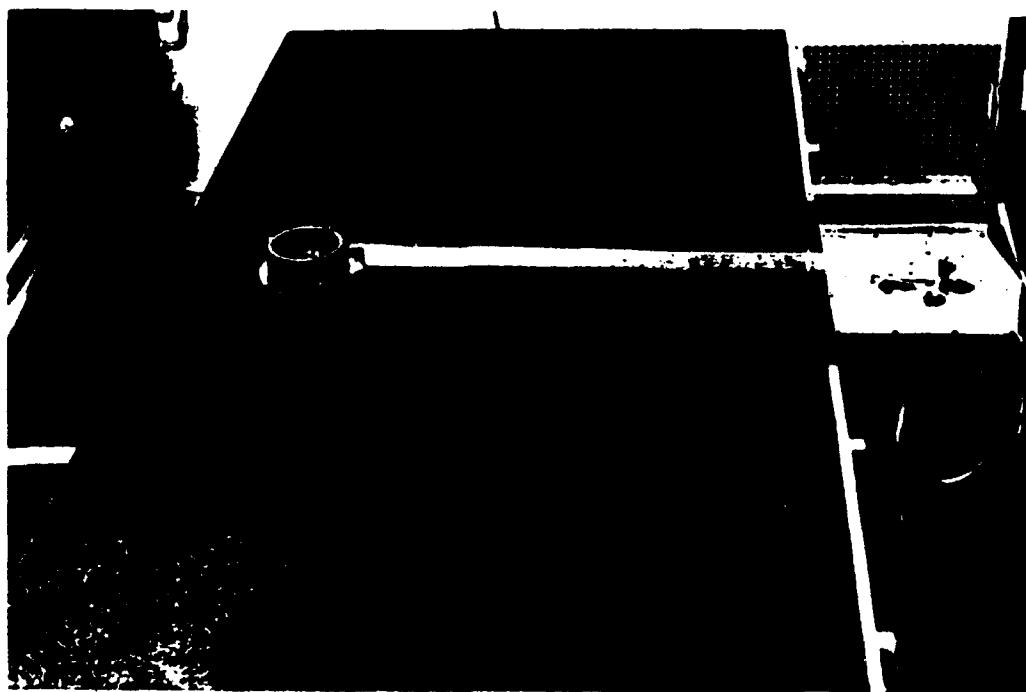
In a paper from the University of Akron [4], a control law was presented in simulation which consisted of a fuzzy logic controller plus a nonlinear effects negotiator for a flexible robot manipulator. In this model the nonlinear effects negotiator was used to cope with the uncertainty of the mathematical model. They used an error response plane

method to obtain fuzzy control rules and investigated the degree of overlap of membership functions for the fuzzy rules. They used a 5x5 rule matrix to obtain these results and sensed both tip displacement and joint angle. Their results showed that the fuzzy logic controller was robust for model uncertainties, nonlinearities, and disturbances.

The objectives of the research conducted for this thesis were to: a) design a fuzzy logic controller to minimize the time for a known, flexible plant and a known payload to slew a given distance, b) verify the robustness with varying payloads and plants, and c) validate the results with hardware-in-the-loop realization of the controller logic. However, unlike the work at the University of Akron, no nonlinear effects negotiator will be used and only joint angle sensing is available. The known plant and payload case performance was first established through the use of a proportional, integral, and differential (PID) controller, a customary approach to robot arm control. Both of the controllers, as well as the varying plants and payloads were implemented in hardware on the University of Washington Control System Laboratory robot arm test facility [6].

### 1.3 System Configuration

Each component of the hardware of the robot arm and equipment used to complete the thesis is described in this section. The general system includes a torque motor to drive the system, a manipulator arm that attaches to the motor, and an endpiece that the arm attaches to. The endpiece is then floated on a large granite table using compressed air, thus minimizing friction. Another benefit of this implementation is that torsional deformations of the arm are minimized.



3

Figure 1.1 Experimental Configuration

### 1.3.1 Torque Motor

A brushless DC torque motor from Aeroflex, part number TQ40W-11P, is the only actuator for the system. The motor has a limited range of sensitivity, but is 100% sensitive for +/- 20 degrees and is 75% sensitive for +/- 60 degrees. The peak torque is 260 inch ounces. The dimensions include an outer diameter of 4 inches, an inner diameter of .375 inches and a width of 1.5 inches. The motor weighs 51 ounces.

### 1.3.2 Manipulator Arm

The original design manipulator arm has a height of .0381 m, a width of .003177 m, and is made of aluminum 6061. This aluminum has a stiffness, E, of  $70 \times 10^9$  N/m<sup>2</sup> and the beam has a moment of inertia of  $1.02 \times 10^{-10}$  m<sup>4</sup>. The actual length is 1.029 m with an effective length of .96583 m.

### 1.3.3 Endpiece

The original design endpiece is a 7 inch diameter, .6 inch thick Plexiglas disk. Three holes, starting with diameters of .6 inches on the bottom and angling into .2 inches at the top, allow compressed air to be expelled in order to allow the disk to float. The secondary endpiece is also a disk of the same dimensions, but it also includes a mass of 192 ounces.

### 1.3.4 Granite Table

A 4 feet by 8 feet by 9.75 inches granite table is the platform for the robot arm. The top has been polished to be smooth and flat.

### 1.3.5 Optical Encoder

A Hewlett Packard HEDS-6010, 3 channel, 56mm, high resolution, incremental, optical encoder is used to read the angular displacement of the arm, theta, from the motor. The encoder consists of an LED source and lens which transmits collimated light from the emitter module through a metal code wheel and phase plate into a detector lens. The shaft contains 1024 cycles per revolution which allows a high level of resolution. A supply voltage of between -0.5 and 7 volts is needed.

### 1.3.6 Quadrature Decoder

The Hewlett Packard HCTL-2020 chip uses a 16-bit counter to perform the quadrature decoder, counter, and bus interface function.

### 1.3.7 Matrix<sub>X</sub>/SystemBuild Software

Computer analysis, simulation, and code generation were performed using Matrix<sub>X</sub> and SystemBuild software by Integrated Systems Inc. (ISI). These linear/non-linear

applications were run on a SPARCstation10 by Sun Electronics. SystemBuild, which uses Matrix<sub>X</sub> as its mathematics platform, performs time simulations as well as generates C code to be used for real time application. SystemBuild includes an interactive animation package which allows inputs and outputs from the system to be examined and changed during real time application.

### 1.3.8 AC100-C30

The AC100-C30 is an ISI hardware/software interface which is embodied in a Dell 466/m computer. When connected to the SPARCstation by ethernet, the AC100-C30 is able to receive, compile, link, and run the C code generated by SystemBuild. If desired, the ethernet is also able to carry the input/output data from the AC100-C30 back to the SPARCstation where it may be observed using the interactive animation. This feature is not required, however, for the system to run. Once the C code has been downloaded to the AC100-C30, the system can run without the presence of the SPARCstation.

## Chapter 2: Model Structure

### 2.1 Overview

In control systems engineering, it is not uncommon for large amounts of time to be wasted due to inexact plant models. After time has been spent creating a simulation, due to inaccuracies in the system model, the control system often will not work on the actual hardware. There are two methods that may be used to avoid this situation; system verification and robust design. If the hardware is known and available, tests may be performed to test the dynamic characteristics and compare the results to the system model. If the exact hardware is either unknown or unavailable, however, this method will not work and a large amount of robustness may have to be designed into the control law.

Because of this plant uncertainty, there has been an increase in research focused on fuzzy rule-based control, based on fuzzy logic theory [4]. After the success of fuzzy logic control in other engineering areas, further work demonstrated the ability of fuzzy logic to control mobile or rigid robots [4]. Finally, research performed at The University of Akron shows in simulation that fuzzy logic could control a flexible manipulator with a large amount of robustness. These results were not validated with hardware-in-the-loop.

### 2.2 Plant Model

Theoretical modeling and verification of the plant model has been accomplished for a 2 link arm by Oakley at Stanford University [5] and for a 1 and 2 link arm by Evers at the University of Washington [1]. This research uses the 1 link arm model developed by Evers. Both Oakley and Evers use a Lumped Spring Mass Damper (LSMD) modeling technique where the distributed effects of elasticity, inertia, and damping are approximated by a finite number of torsional springs, dampers, link elements, hub elements, and tip elements. The inertial and kinematic parameters of the manipulator elements were developed analytically and used in the mathematical model. Damping was determined experimentally and inserted back into the analytical model.

There are 6 assumptions used in the model:

1. In plane bending and motion
2. Ideal pinned joint action
3. No frictional effects
4. Small in plane deflection
5. Ideal linear amplifier characteristics
6. Ideal non-linear motor characteristics [1]

Using these assumptions, non-linear dynamic models were developed for the two link manipulator using the Lagrange method for low order models. The resulting non-linear models were then adapted, by Evers, to the single link arm. The non-linear models for the single link arm are of the form:

$$M(z)\dot{u} + V(z, u) + R\dot{z} + Kz = T\tau \quad (2.1)$$

$$u_i = \dot{\Theta}_i + \sum_{k=1}^{i-1} \dot{q}_{ik} \quad (2.2)$$

$$u = [u_1, \dots, u_{n_1}]^T \quad (2.3)$$

$$z = [\Theta_1, q_{11}, \dots, q_{1n_1}]^T \quad (2.4)$$

$$\tau = \tau_1 \quad (2.5)$$

where  $M$  contains the mass and inertia manipulator properties,  $V$  contains the Coriolis and centripetal terms,  $R$  contains the coulomb torsional damping,  $K$  represents the torsional stiffness,  $T$  represents the torque distribution matrix, and  $\tau$  represents the input torque. The remaining physical properties are defined in Figure 2.1.

These non-linear equations were then linearized using Taylor's expansion. The resulting linear equation of motion is of the form:

$$\tilde{M}\Delta\ddot{z} + R\Delta\dot{z} + K\Delta z = T\Delta\tau \quad (2.6)$$

The result of the plant representation is a state space system with a single input and 42 outputs.  $x$  represents the states,  $\Delta\tau$  the input, and  $y$  the outputs. The outputs consist of the 40 states as well as 2 local coordinates which are not used in this research. The input is the torque generated by the control law. The linear model in standard state space convention is:

$$\begin{bmatrix} \tilde{M} & 0 \\ 0 & I \end{bmatrix} \frac{d}{dt} \begin{Bmatrix} \Delta\dot{z} \\ \Delta z \end{Bmatrix} = \begin{bmatrix} -R & -K \\ I & 0 \end{bmatrix} \begin{Bmatrix} \Delta\dot{z} \\ \Delta z \end{Bmatrix} + \begin{bmatrix} T \\ 0 \end{bmatrix} \Delta\tau \quad (2.7)$$

$$\frac{d}{dt} \begin{Bmatrix} \Delta \dot{z} \\ \Delta z \end{Bmatrix} = \begin{bmatrix} \tilde{M} & 0 \\ 0 & I \end{bmatrix}^{-1} \begin{bmatrix} -R & -K \\ I & 0 \end{bmatrix} \begin{Bmatrix} \Delta \dot{z} \\ \Delta z \end{Bmatrix} + \begin{bmatrix} \tilde{M} & 0 \\ 0 & I \end{bmatrix}^{-1} \begin{bmatrix} T \\ 0 \end{bmatrix} \Delta \tau \quad (2.8)$$

$$\dot{x} = Ax + B\Delta\tau \quad (2.9)$$

$$y = Cx + D\Delta\tau \quad (2.10)$$

$$A = \begin{bmatrix} \tilde{M} & 0 \\ 0 & I \end{bmatrix}^{-1} \begin{bmatrix} -R & -K \\ I & 0 \end{bmatrix} \quad (2.11)$$

$$B = \begin{bmatrix} \tilde{M} & 0 \\ 0 & I \end{bmatrix}^{-1} \begin{bmatrix} T \\ 0 \end{bmatrix} \quad (2.12)$$

$$x = [\Delta \dot{z} \quad \Delta z]^T = \begin{pmatrix} \dot{\theta} \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_{(n-1)} \\ \theta \\ q_1 \\ q_2 \\ q_{(n-1)} \end{pmatrix} \quad (2.13)$$

torque motor

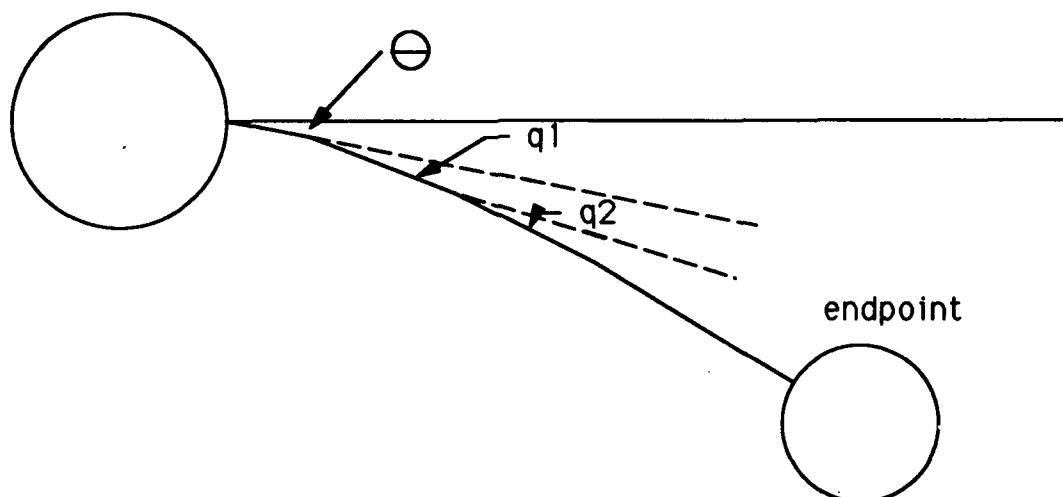


Figure 2.1 System State Description

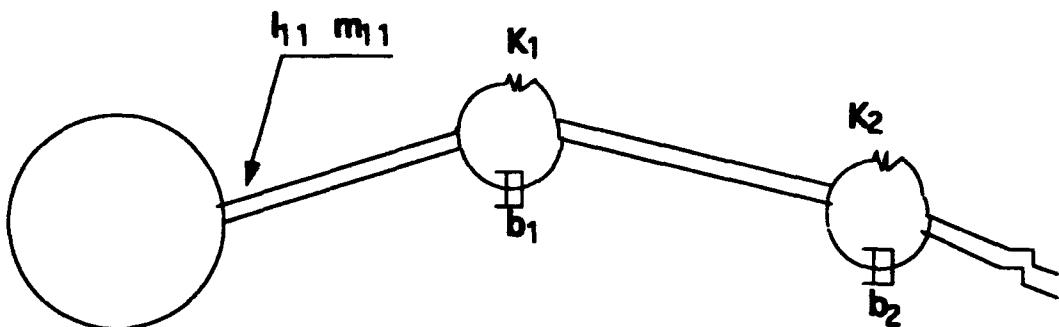


Figure 2.2 Finite Element Representation

In the model, the manipulator arm may be sectioned into as many finite elements as the user deems necessary to give an accurate response of the important modes (Figure 2.2). Model reduction may then be used to obtain an accurate lower order model. In the work by Evers it was concluded that only the first 3 modes would be important for the system and that a finite element grid of 20 members would accurately represent these 3 modes. For a 20 member bar this results in 40 states. The states of the system, shown in Figure 2.1, include the angle at the shoulder in radians, the velocity of this angle change in radians per second, the distances that each of the finite element members have displaced due to the flexibility in the arm in radians, and the velocity of these displacements in radians per second. The eigenvalues of the resulting system show that the system has 38 stable poles and 2 neutrally stable complex poles. The first twelve modes, corresponding to the first 25 poles, are shown in Figure 2.3. The first eight of these modes are at 12 Hz, 37 Hz, 73 Hz, 101 Hz, 102 Hz, 103 Hz, 106 Hz, 112 Hz, 128 Hz, and 154 Hz. The remaining 15 poles are all stable and sit on the real axis as far out as 144,000 Hz.

The computer program used to generate the plant model, the derivation of the mathematical model, a listing of poles and zeros, as well as bode and root locus diagrams are shown in the appendices.

### 2.3 Model Reduction

The state space model developed from the 20 piece finite element grid results in a 40 by 40 matrix. Although this provides an accurate representation of the actual system, it is undesirable to use a system this large for control design studies due to the inefficiencies caused by the large number of calculations required for a matrix of this size. To make the design process more efficient, the state space model is reduced to 10 states.

There are many methods that are available for model reduction, however, most of these methods have a restriction that the system be completely stable. Since the robot arm contains a pair of neutrally stable poles these methods are not immediately available for use. To bypass this problem the state matrix is partitioned to separate the stable poles from the nonstable poles. By doing this, the stable matrix may be reduced, then the unstable part of the matrix may be added back on.

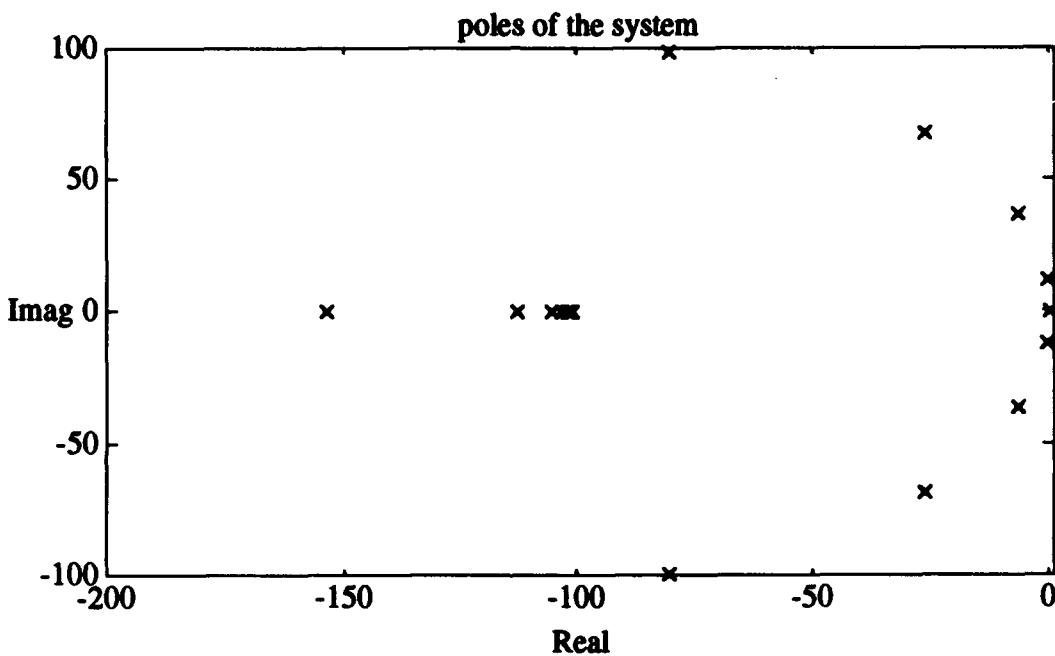


Figure 2.3 Open Loop Poles of the 40 State System

The reduction method used for this system is the truncation of a balanced realization. A balanced realization balances the ease of control against the ease of observation. To do this the singular value decompositions of the observability grammian,  $Q$ , and the controllability grammian,  $P$ , are combined to result in a single singular value decomposition. This is then used to form a transformation for the system with balanced observability and controllability. Once the system is balanced, the smallest, and, therefore, hardest to observe or control, eigenvalues are eliminated. This leaves a smaller system with only the hardest to control and observe states missing. This can cause a problem, however, if the hard to control or observe states happen to be the states of importance. The remaining 10 poles of the reduced system are shown in Figure 2.4. The

poles at the origin remain unchanged, while the remaining 8 poles have moved off of the real axis and have become 4 sets of complex poles. The flexible modes that remain are at 37 Hz, 111 Hz, 215 Hz, and 370 Hz. Although some of the modes from the full model have been removed, the second flexible mode at 37 Hz remains. From experimentation, it is this mode that is most easily excited. Thus the mode that is the most relevant to implementation remains in the model.

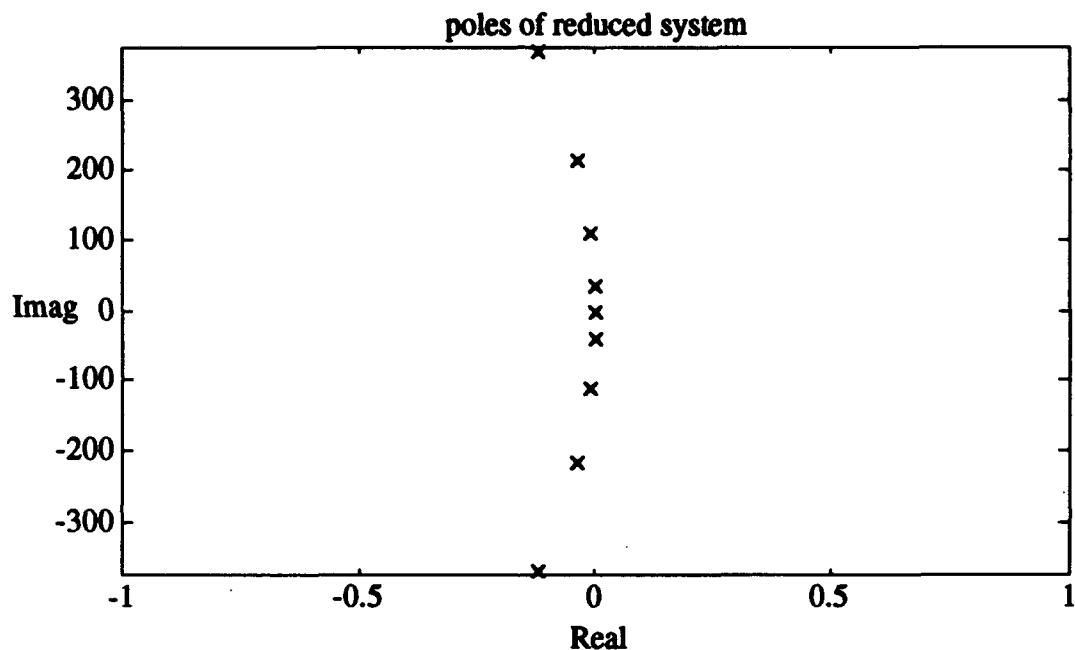


Figure 2.4 Open Loop Poles of Reduced Model

#### 2.4 Digitization

Once the system is reduced, the state space is transformed from continuous to digital form. The Tustin's method is used with a sampling time of .01 seconds.

## Chapter 3: Tests and Results

### 3.1 Overview

The system model as well as the hardware and software described in Chapter 2 were new to the University of Washington System Controls Laboratory. Because of the inherent problems associated with any new hardware or software system, a well understood test model was first used to validate the system model and ensure proper connections between the software and the hardware. To accomplish this objective a proportional, integral, and derivative (PID) controller was developed. After this was done the fuzzy logic controller was developed. The fuzzy logic controller was designed and first tested using the verified plant model, but was then tested on other unknown plants.

### 3.2 Feedback System

Both controllers were developed using negative feedback of the state  $\theta$  from the plant.

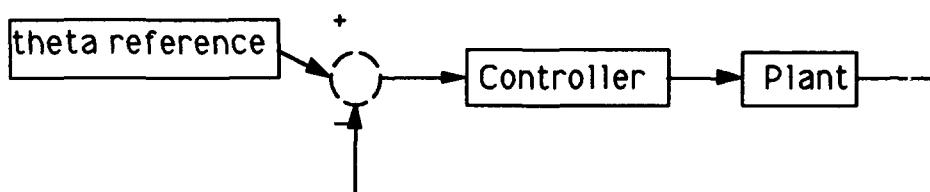


Figure 3.1 Feedback Configuration

When slewing a flexible arm there is a trade off between two slewing techniques. The first possibility is to slew the arm as fast as possible with a step input. Assuming that the motor is able to accommodate this command, the rapid onset of the movement will most likely excite the flexible modes of the arm. These extra dynamics will increase the work load of the controller and may increase the time to settle at the desired location. The other possibility is to slew the arm very slowly as to not excite any flexible modes of the arm. This is not a hard control task, but the response is also very slow. To maximize the advantages of both possibilities while easing the strain of discontinuities on the motor,

the reference theta used to slew the flexible arm is a smooth ramp (Figure 3.2) used by Oakley [5]. The equation for the reference input is <sup>13</sup>

$$\Theta_{ref} = [\Theta_{final} - \Theta_{initial}] [6T^5 - 15T^4 + 10T^3] + \Theta_{initial} \quad (3.1)$$

$$T = \frac{t}{T_{slew}} \quad (3.2)$$

$T_{slew}$  is the time allowed for the reference theta to move from the initial position to the final position.

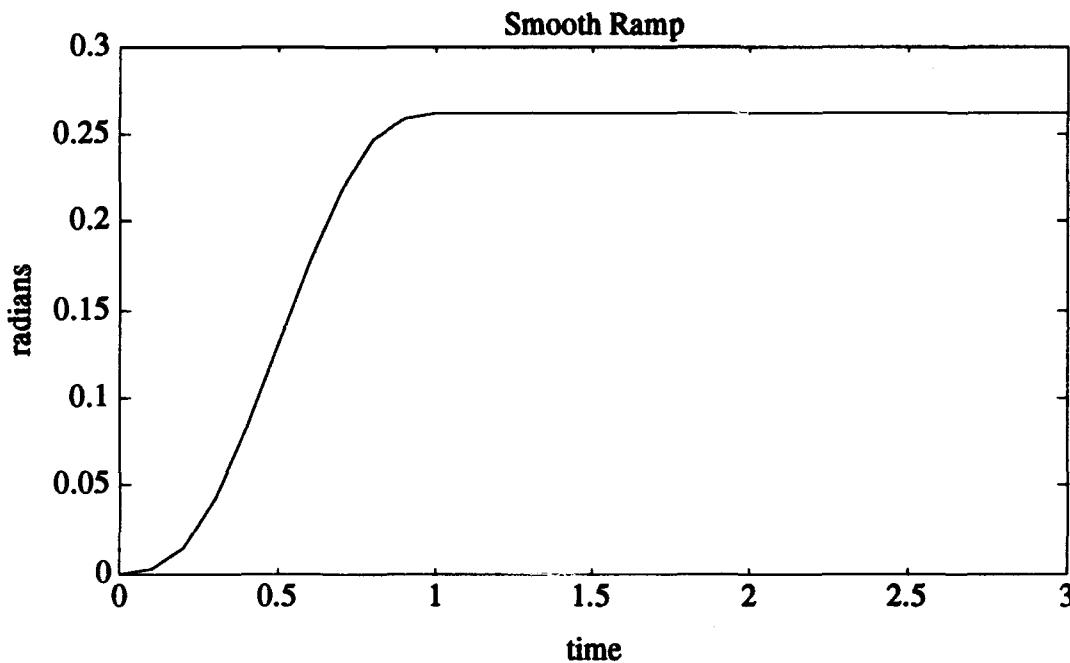


Figure 3.2 Smooth Ramp Reference Theta to 15 Degrees,  $T_{slew}=1$  sec

### 3.3 PID Controller in Simulation

The PID controller for the single link manipulator was designed as a digital system using root locus and time response to determine the gains.

Table 3.1 PID Gains

PID Gains	
Proportional	0.3
Integral	0.1
Derivative	0.6

These gains provide a controller with a smooth torque output (Newton-meters converted to volts as 1.9208 volts per Nm) that in simulation is able to slew the arm to 10 degrees with a steady state error of .3 percent, a rise time of .58 seconds, and a 5 percent settling time of 4.5 seconds. The overshoot for this system is 16 percent.

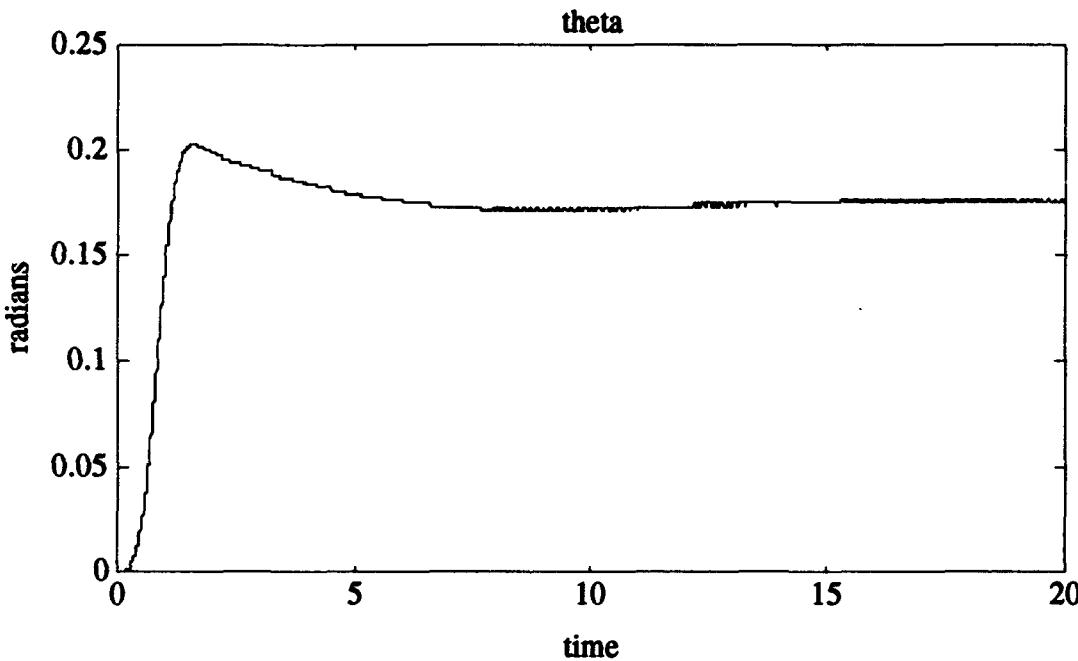


Figure 3.3 PID Simulation Theta 10 Degree Slew

For a 15 degree slew these numbers change to a steady state error of 0.7 percent, a rise time of 0.58 seconds, a settling time of 4.5 seconds, and a 15 percent overshoot. The 'chatter' in both the torque and theta response is a result of the quantization of the system. If the test is run without simulating the hardware interface, this result is not observed.

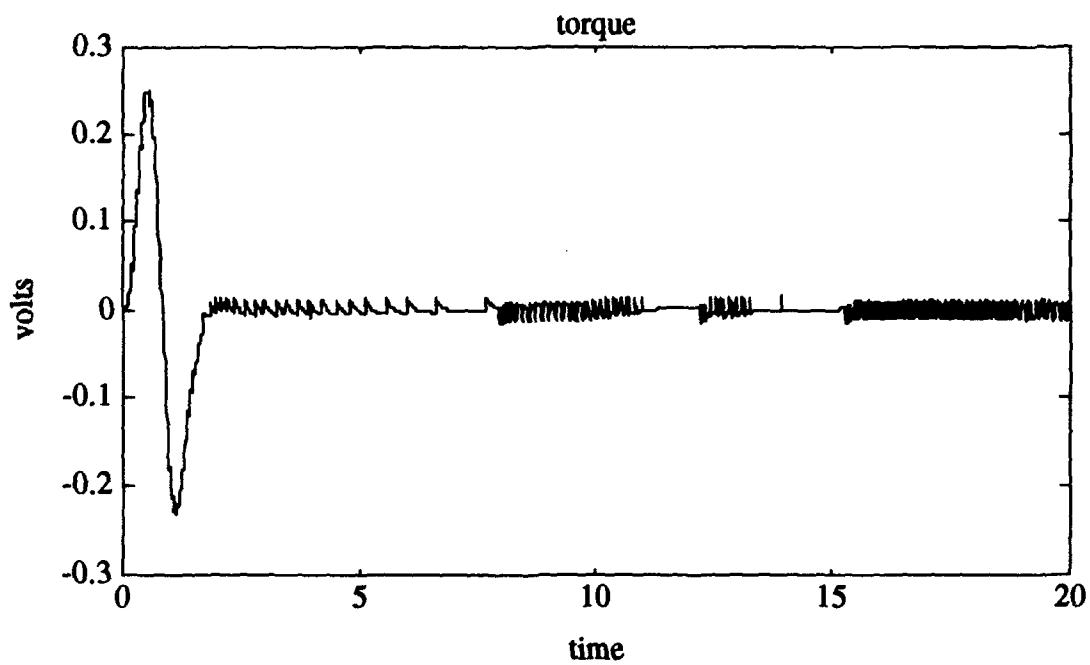


Figure 3.4 PID Simulation Torque 10 Degree Slew

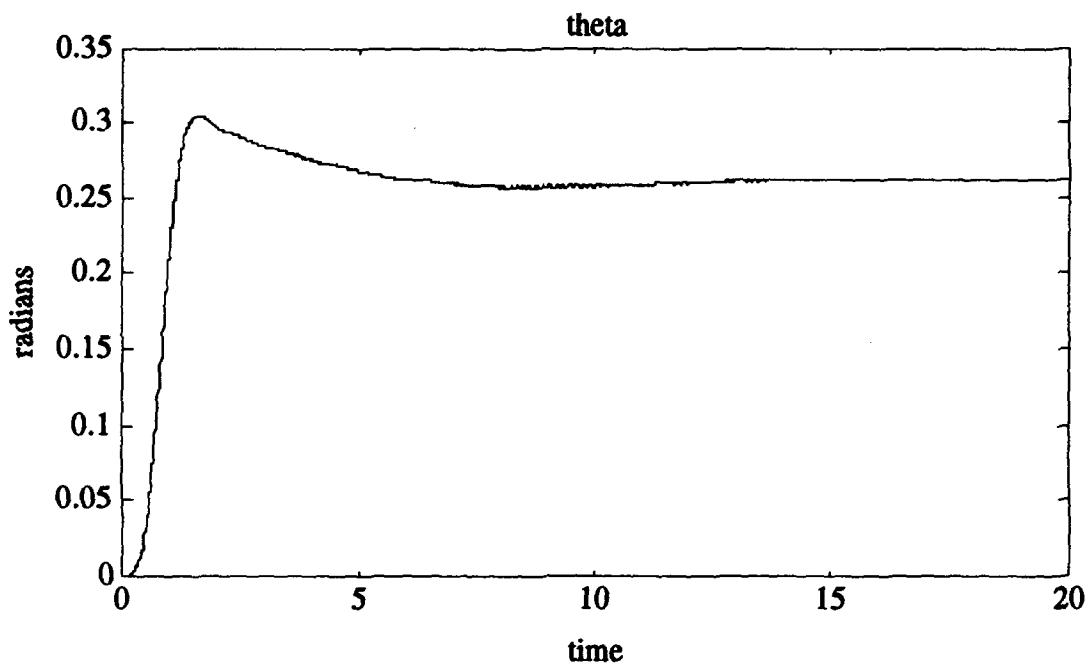


Figure 3.5 PID Simulation Theta 15 Degree Slew

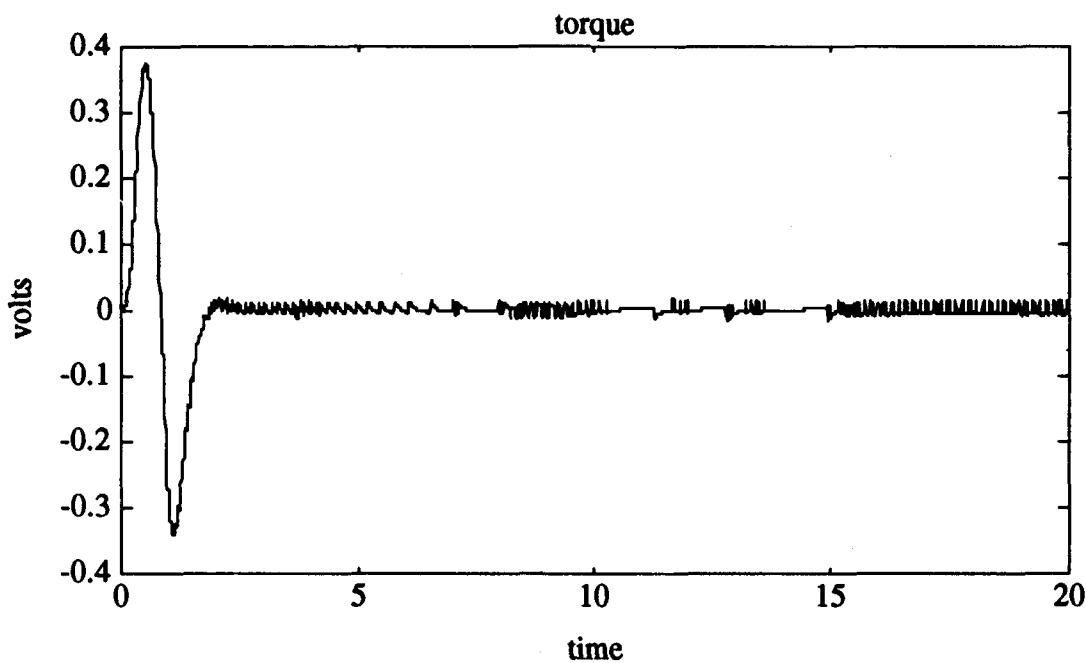


Figure 3.6 PID Simulation Torque 15 Degree Slew

### 3.4 PID Controller In Hardware

Using the PID controller on the hardware gave similar results to those obtained in the simulation. For a 10 degree slew the steady state error increased from 0.3 percent to 5.73 percent and the rise time rose from 0.58 seconds to 1.55 seconds. The overshoot, however, decreased from 16 percent to 10.57 percent and the settling time, accounting for an initial displacement of the manipulator arm rose from 4.5 seconds to 8 seconds. The comparisons are similar for the 15 degree slew. The steady state error increased from .7 percent to 2.7 percent, the rise time increased from 0.58 seconds to 2.81 seconds, and the settling time increased from 4.5 seconds to 7.5 seconds. As with the 10 degree slew case, however, the overshoot decreased from 15 percent to 5.71 percent.

For all of the performance criteria except the percent overshoot, the result is not surprising. In general, some degradation in performance from simulation to implementation is expected. The reduction in percent overshoot may be attributed to a combination of the modeling of damping and stiffness of the manipulator. These values in the model have been determined experimentally and are subject to being updated.

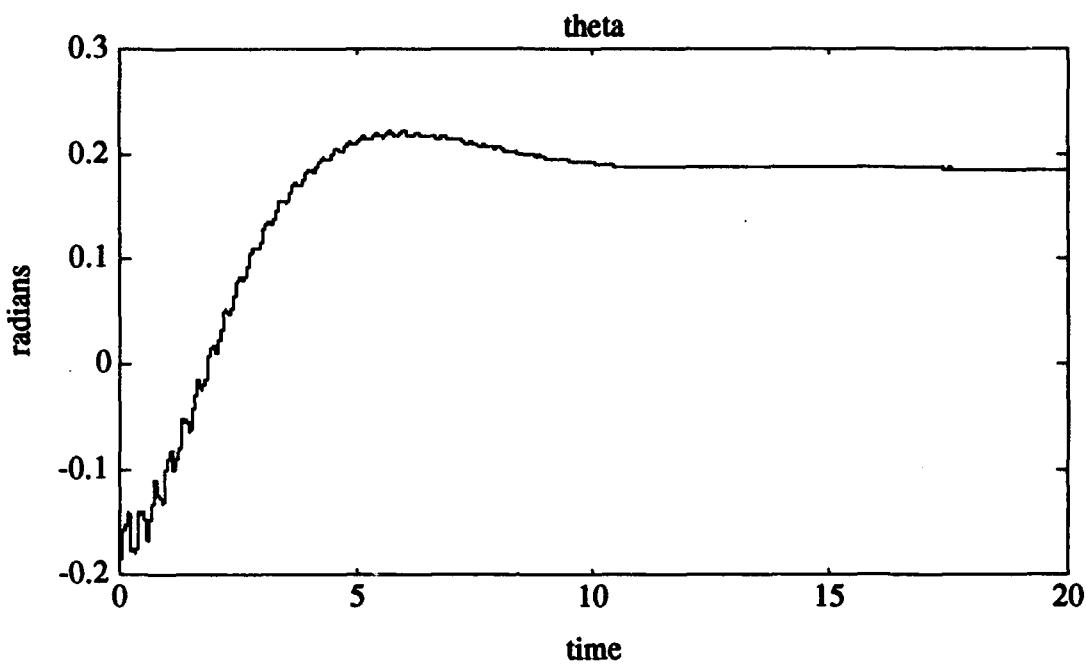


Figure 3.7 PID Hardware Theta 10 Degree Slew

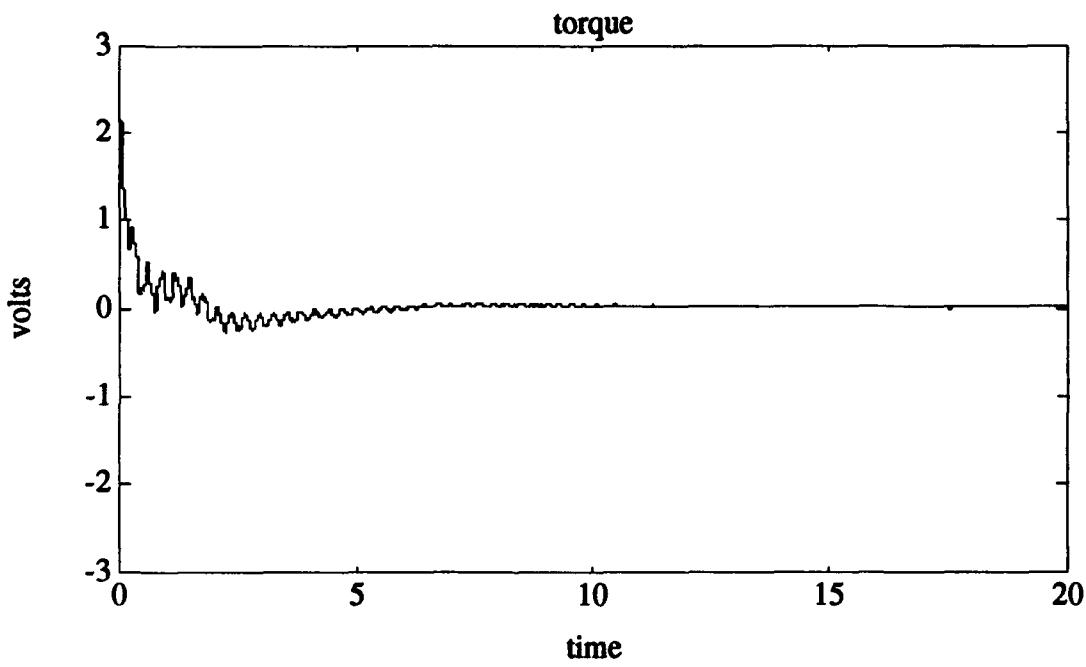


Figure 3.8 PID Hardware Torque 10 Degree Slew

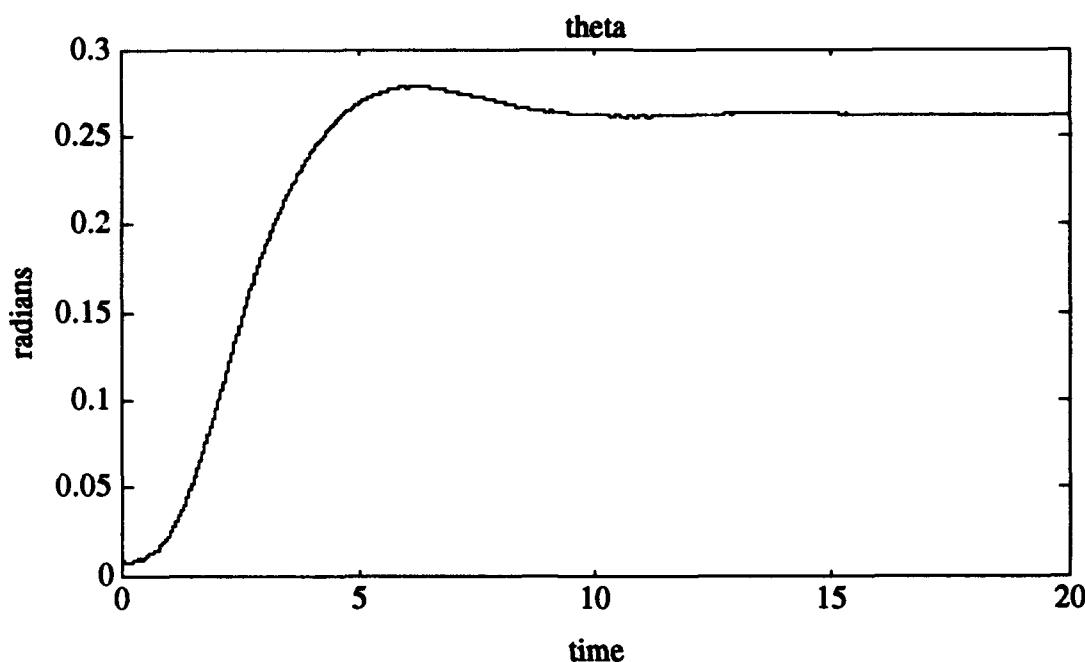


Figure 3.9 PID Hardware Theta 15 Degree Slew

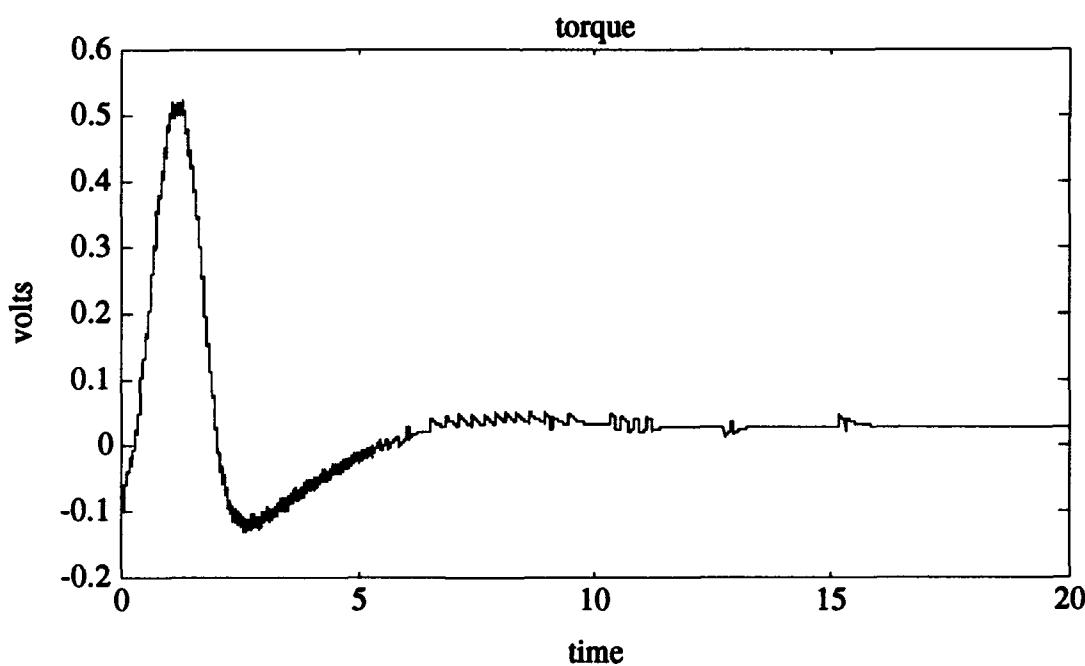


Figure 3.10 PID Hardware Torque 15 Degree Slew

### 3.5 Fuzzy Logic Controller Theory

A fuzzy logic controller is a 5-step, rule based system that is able to combine in parallel an array of inputs, and develop a single output that is unique to the specific inputs. These inputs are based on qualitative information and then are connected with each other using a preset 'knowledge base.' The input is then evaluated against the 'fuzzy data' using 'membership curves' to determine the appropriate output.

A fuzzy logic system contains no true states, but consists of a database containing definitions of the inputs and outputs. The database includes the type of data, the range of acceptable values, and the membership functions to be applied to the data. The type of data refers to crisp versus fuzzy which is best explained by example. When considering a summer day, people have varying opinions as to the temperature when the day becomes 'hot.' In this scenario the actual temperature is the crisp value and the term 'hot' is the fuzzy data. Whereas almost nobody would say that 50 degrees F is hot, and almost everyone would say 100 degrees F is hot, there remains a region in between where different people would express different opinions. Therefore, the phrase 'it is hot' represents a fuzzy variable. The range of values describes the relevant values of the crisp data. For temperature, the universe of acceptable values may range from 0 to 100 degrees. Anything outside this limit would not be of concern and could be considered to be either the high or low limit for the fuzzy application.

The knowledge base is the programmable, control aspect of a fuzzy logic controller. The knowledge base is composed of If-Then rules which can be interpreted in the fuzzy logic process. Both the condition, If, and the assertion, Then, use crisp variables put into a fuzzy context. For example, temperature and humidity are crisp variables, but a rule may be If temperature is hot (hot converts temperature into a fuzzy context), and humidity is musty, Then air conditioner power is high. The database and knowledge base combine to form the information needed to run a fuzzy logic based controller.

The first of the five steps is that of fuzzification. This step simply converts a input, or a crisp value, into fuzzy data. The curve that defines the relationship between the crisp data and the fuzzy variable is called the membership curve.

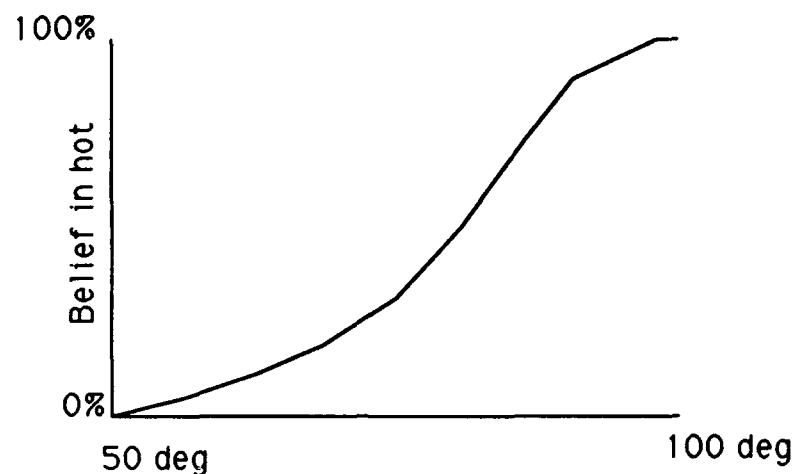


Figure 3.11 Membership Curve Concept

This membership curve is used to fuzzify the crisp value of degrees into the fuzzy variable hot. With the representation of  $hot = y(x)$ , qualifiers such as somewhat hot =  $\sqrt{y(x)}$  or very hot =  $y(x)^2$  may be used to sketch related curves.

For any input there may be multiple membership curves. Temperature, for example, may have 3 curves: cold, lukewarm, and hot (Figure 3.12). These curves are overlapped so that for any temperature there will always be at least a 50% belief in any one curve. By overlapping curves, the difference of opinion shown in the above membership curve may be used to enhance the control. For a temperature of 70 degrees there may be a 75% belief in hot and a 25% belief in lukewarm and a 0% belief in cold.

Once the variables have been fuzzified, combinational logic is used to find a belief value in the desired output. Normally there are 2 inputs that are being considered for the fuzzy rule. The beliefs for each input are then compared using the fuzzy conjunction of minimum. For example, for a temperature of 70 degrees and a humidity of 75%, the belief in hot may be 50% and the belief in humid may be 70%. In this case, the output, perhaps air conditioner power, would be medium to a degree of 50%. However, as shown in Table 3.2, there are more possible combinations of relevant membership functions that must be considered. Once the minimum operator has been used to develop the belief for each specific output, the maximum operator is used to combine the results of any rules that call for the same action. This gives the final fuzzy output as 'Power is Medium to a degree of .50 and Low to a degree of .50.'

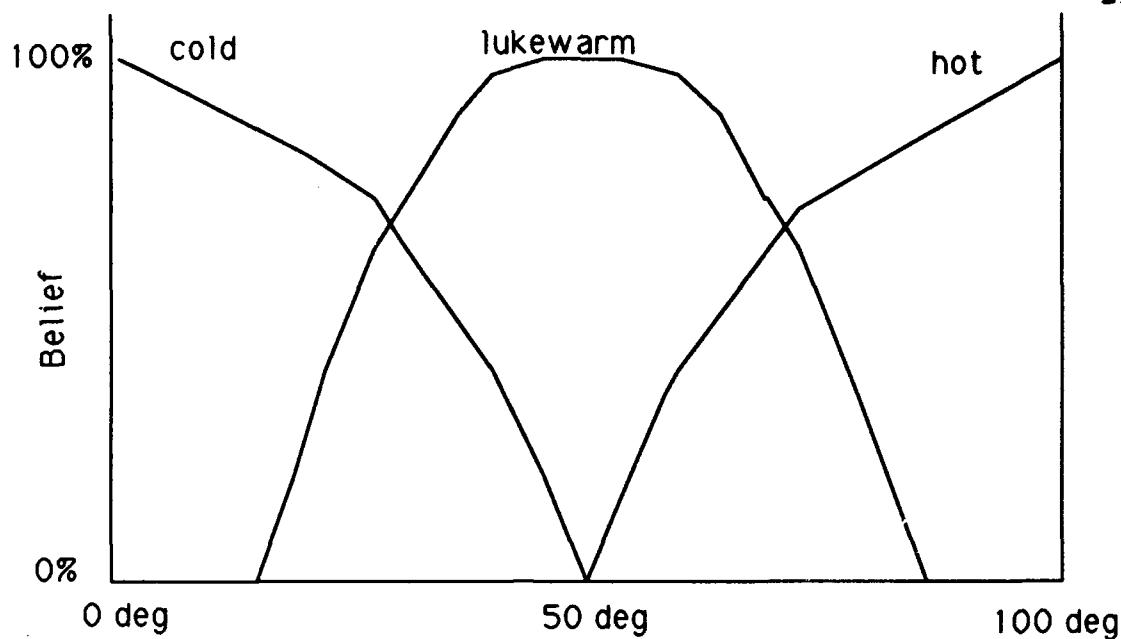


Figure 3.12 Multiple Membership Curves

Table 3.2 Conjunction of Minimum

Temperature	Degree	Humidity	Degree	Power	Degree
Hot	0.5	Musty	0.7	Medium	0.5
Lukewarm	0.5	Average	0.3	Low	0.3
Hot	0.5	Average	0.3	Medium	0.3
Lukewarm	0.5	Musty	0.7	Low	0.5

The third step, implication, is an intermediate stage in reaching a crisp output from the fuzzy data. Implication 'states the way a fuzzy curve is implicated or deduced from the belief value obtained in the rule condition.' [2] There are two common techniques used for this action: the Mamdani method, which uses the minimum operator, and the Larson method, which uses Boolean conjecture. For the purpose of this research, the Mamdani method is used. This method takes the minimum of the belief in the 'If' condition, and the class membership curve in the assertion  $[Min(c, y(x))]$ . [2] The result of this is a membership curve that has been leveled off at the appropriate belief value.

The fourth step, defuzzification, converts an implicated fuzzy membership curve into a crisp output. As with implication, there are several methods with which

defuzzification may be accomplished. The method used in this research is the centroid method. The centroid method splits the area under the curve into two parts and chooses the mean value for the output using

$$\frac{\int x \cdot y(x) dx}{\int y(x) dx} \quad (3.3)$$

In Figure 3.13, with area A equaling area B, this would result in a crisp output of 650.

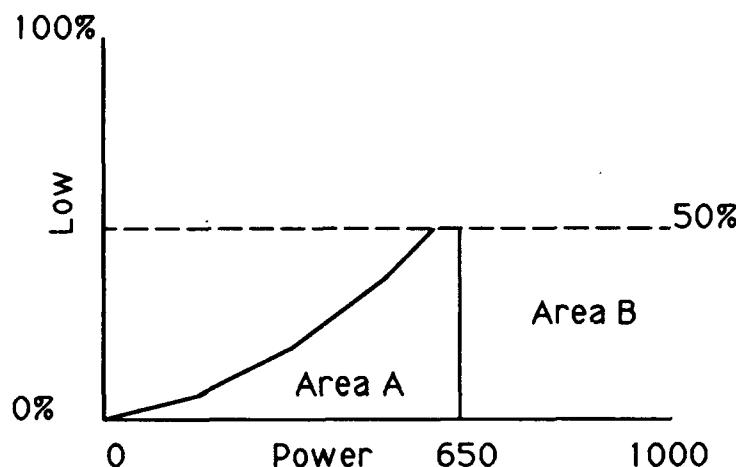


Figure 3.13 Implication

Since there is more than one fuzzy output from the combinational logic, and therefore more than one implicated membership curve and defuzzified output, there must be a way to convert these multiple outputs into a single output for the controller. This is accomplished in the final fifth step, aggregation. Aggregation uses the average of all of the crisp outputs to determine this one command. Continuing the temperature example, there are two crisp outputs, 650 to a degree of .5, as found by Figure 3.13, and 1000 to a degree of .5, as found in a similar manner. The final crisp output would then be

$$\frac{.5 * 650 + .5 * 1000}{.5 + .5} = 825 \quad (3.4)$$

### 3.6 Fuzzy Controller

Four separate fuzzy controllers were designed and evaluated during this research. The first two use  $5 \times 5$  control matrices while the last two use  $7 \times 7$  matrices. All four control laws use 2 inputs:

$$e = \Theta_{\text{Reference}} - \Theta_{\text{Actual}}$$

$$\dot{e} = \frac{de}{dt} \quad (3.5)$$

to develop a single torque output using the same negative theta feedback as the PID controller.

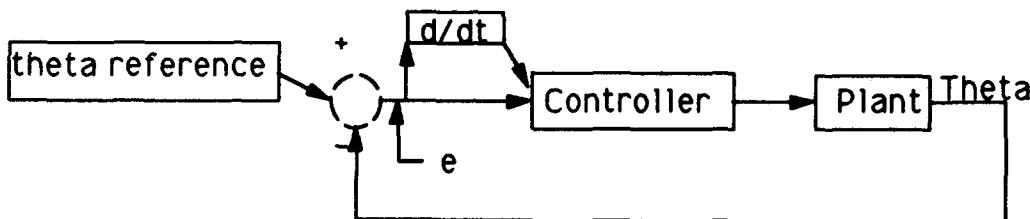


Figure 3.14 Fuzzy Feedback

#### 3.6.1 Five by Five Control Matrix

The first controller developed uses a  $5 \times 5$  control matrix. Angle error,  $e$ , angle error velocity,  $\dot{e}$ , and torque are all broken into 5 divisions: positive medium, positive small, zero, negative small, and negative medium. Torque and  $\dot{e}$  share the same membership curve equations (Table 3.3 and Figure 3.15) for these divisions, while  $e$  uses different curve equations (Table 3.4 and Figure 3.16). Angle error uses these different membership curves to reduce steady state error. By changing the zero membership curve to become more slender, the region where the control law assumes that the angle error is zero becomes more confined. This is not done for the angle error velocity and torque in order to help the system settle. As the angle error gets smaller it is desirable for the system to not react as strongly as if the angle error was large. By increasing the range of zero for the error velocity, the system is not as sensitive to small movements and more sensitive to angle error. By having a large zero range for torque the outputs are more likely to settle as the errors get smaller. These divisions cover the range between -.14 and

.14 radians for  $e$ , -.7 and .7 radians/second for  $\dot{e}$ , and -.38 and .38 volts for torque. Although some values of  $e$  and  $\dot{e}$  will be outside of these zones, the computer is able to use these data by assigning them to the stated positive or negative extreme. Since torque is the output there will not be any voltages outside of the given range. The rules that form the matrix (Table 3.5) have equal weights and are in the form IF  $e$  IS NEGATIVE MEDIUM AND  $\dot{e}$  IS NEGATIVE SMALL, THEN TORQUE IS NEGATIVE MEDIUM.

Table 3.3 Angle Error Velocity and Torque Five by Five Membership Curve Equations

Membership Curve Equations	
POS_MEDIUM	$\text{SIN}(\text{Pi}*(X-0.5))^5$
POS_SMALL	$\text{SIN}(\text{Pi}*(X-0.25))^5$
ZERO	$\text{SIN}(\text{Pi}*\text{X})^3$
NEG_SMALL	$\text{SIN}(\text{Pi}*(X+0.25))^5$
NEG_MEDIUM	$\text{SIN}(\text{Pi}*(X+0.5))^5$

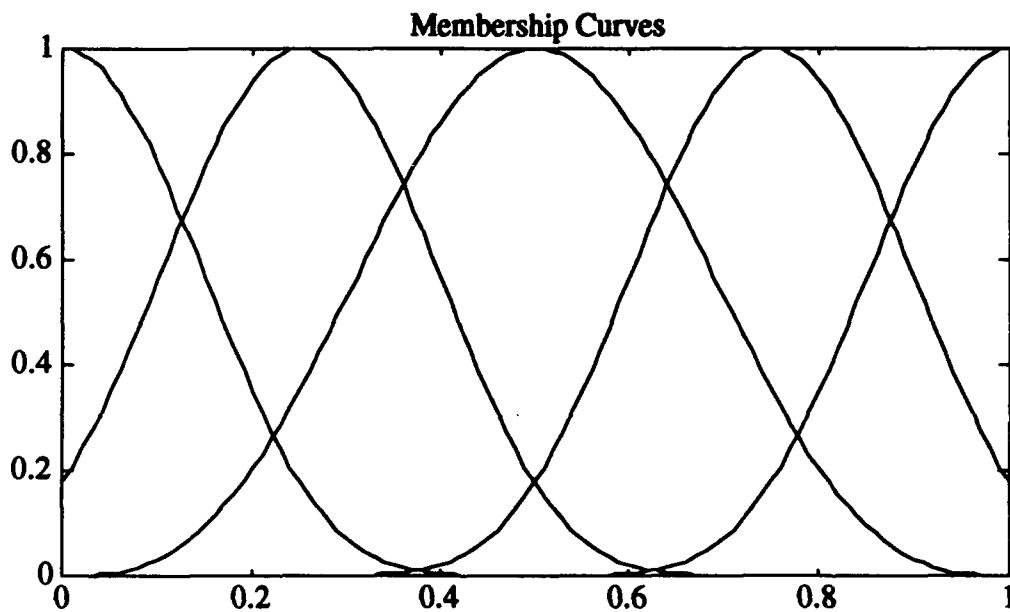


Figure 3.15 Membership Curve Shapes for Angle Error Velocity and Torque

Table 3.4 Angle Error Five by Five Membership Curve Equations

Membership Curve Equations	
POS_MEDIUM	$\text{SIN}(\text{Pi}^*(\text{X}-0.5))^5$
POS_SMALL	$\text{SIN}(1.1*\text{Pi}^*(\text{X}-.254))^5$
ZERO	$\text{SIN}(3*\text{Pi}^*(\text{X}+0.33))^5$
NEG_SMALL	$\text{SIN}(1.1*\text{Pi}^*(\text{X}+0.16))^5$
NEG_MEDIUM	$\text{SIN}(\text{Pi}^*(\text{X}+0.5))^5$

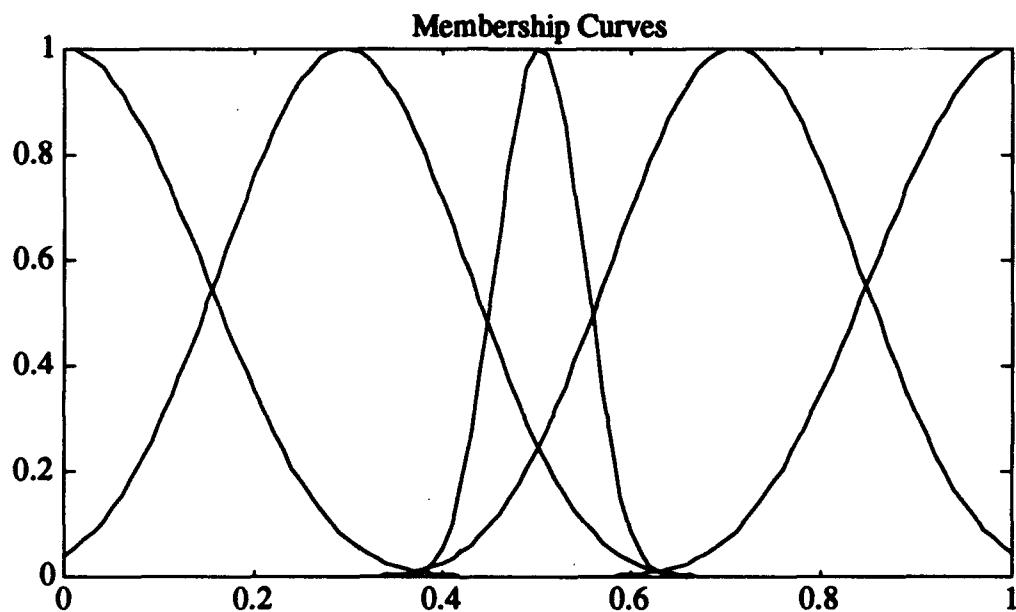


Figure 3.16 Membership Curve Shapes for Angle Error

Table 3.5 Single Five by Five Controller Rule Matrix

	NM	NS	ZR	PS	PM
NM	NM		NM		
NS		NM	ZR	PS	
ZR		NM	ZR	PM	
PS		NS	ZR	PM	
PM			PM		PM

Using symmetry, the computer is able to compensate for unfilled matrix squares so that the entire matrix does not need to be filled.

This controller produces simulations that in general perform poorly. The torque curves are not smooth and the steady state error is large. For a 15 degree slew, .2618 radians, the final value is .2130 radians, or a 18.4% steady state error. The system does not get to within 5% of the desired steady state value, but the steady state settling time is 11 seconds. The one good performance result is a 0.5 second rise time. Unfortunately because of an initial displacement in the wrong direction this does not occur until 2.58 seconds into the simulation. Also, this rise time results in a 78% overshoot.

Fortunately this is not the end of the fuzzy control story. Although this  $5 \times 5$  control law works poorly in simulation, there are two redeeming features. The first, which will be shown later in the paper, is that as the control law matrix expands, the system performance improves. The second is that in hardware the torque gets filtered somewhat by the dynamics of the motor and the electronic amplifier. This in turn takes some of the shake out of the theta response. Also, because fuzzy logic controllers are an iterative process, a change in the torque response early in the slew greatly effects the rest of the slew. The effects of this filtering are seen in the theta and torque responses from a test on the actual hardware.

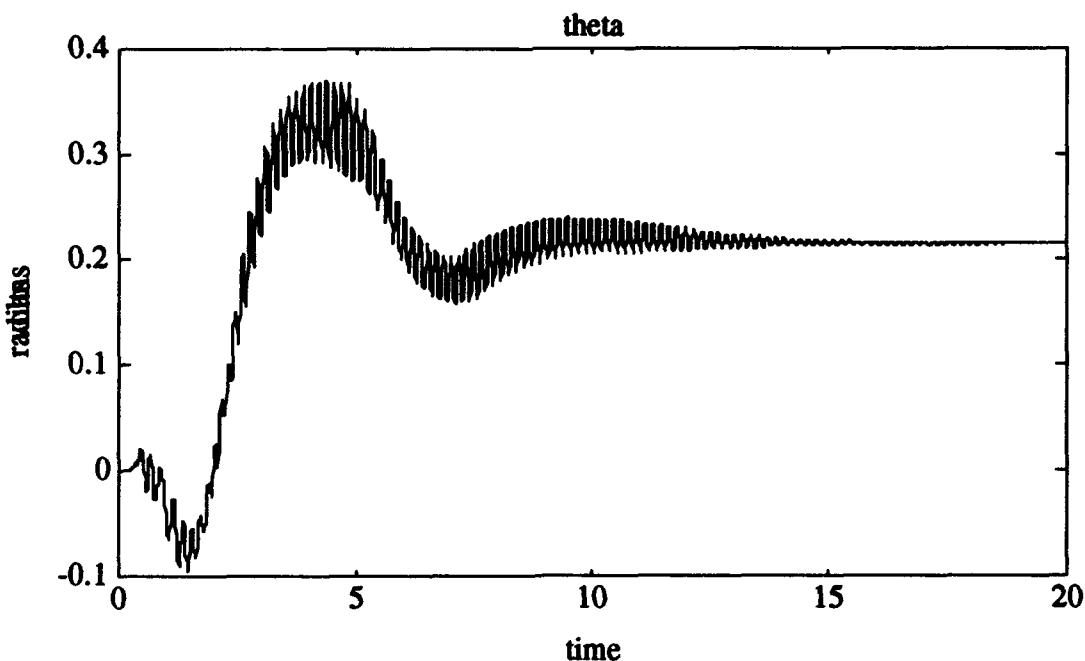


Figure 3.17 Five by Five Theta Simulation, 15 Degree Slew

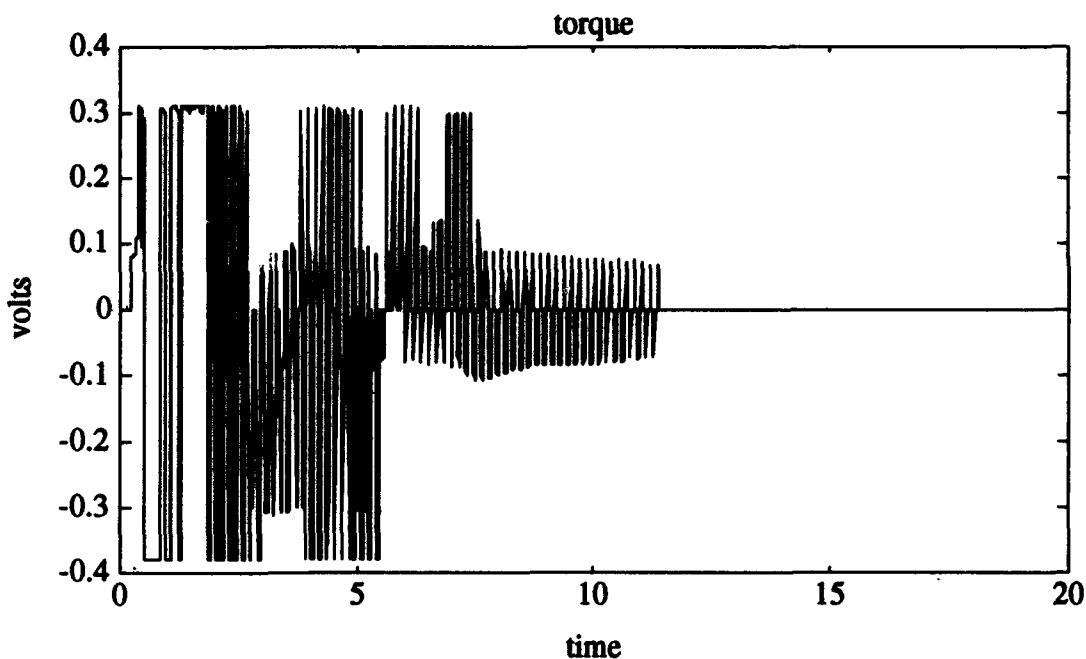


Figure 3.18 Five by Five Torque Simulation, 15 Degree Slew

For a 15 degree slew on the hardware (Figure 3.19 and Figure 3.20), the steady state value is .27 radians. This is only a 3.1% steady state error. The system is a little slower with a 14 second settling time and a 1.142 second rise time. With the slower response, the overshoot also decreases to 52%.

Although the performance values differ between the simulation and hardware results, correlation is maintained in the same basic shape and response. The hardware response does not begin in the opposite direction as the simulation does, but it does resemble the simulation response for the first few seconds. Both results also have 3 peaks reached at approximately the same times. The first peak is at about 4.5 seconds, the second peak is at about 7 seconds, and the third peak is at about 10 seconds.

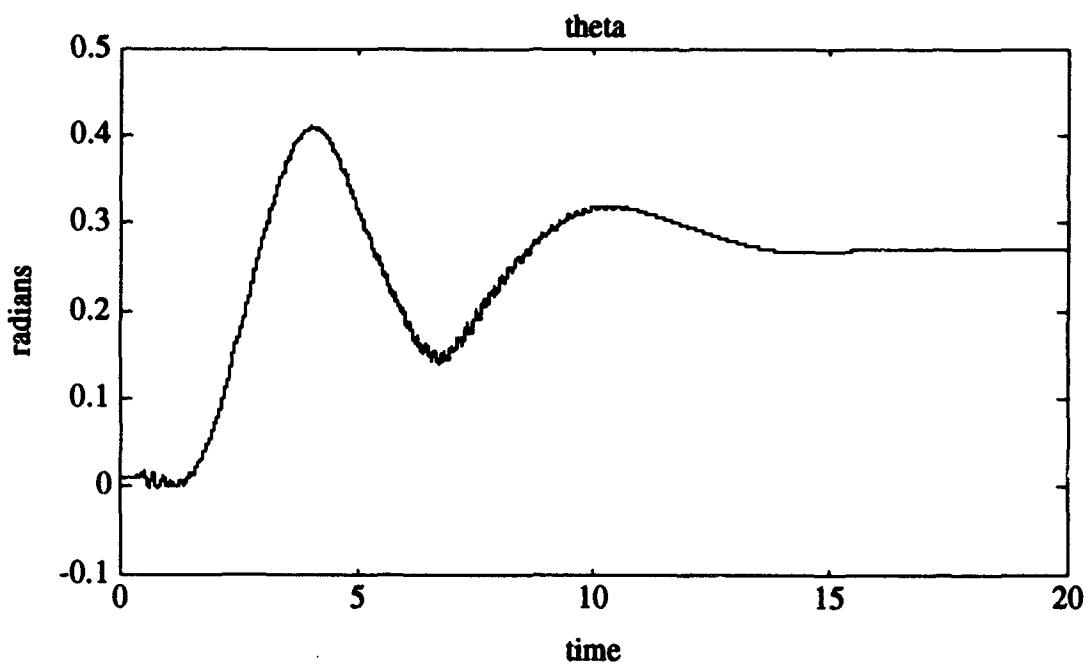


Figure 3.19 Five by Five Theta in Hardware, 15 degree Slew

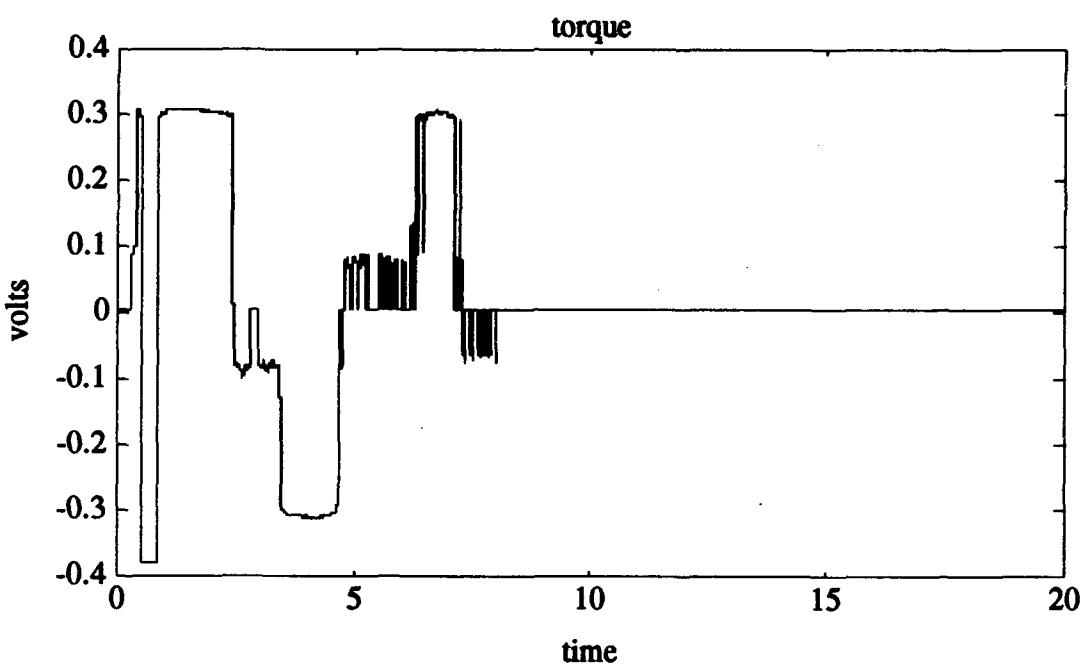


Figure 3.20 Five by Five Torque in Hardware, 15 Degree Slew

### 3.6.2 Two Five by Five Control Matrices

When designing a fuzzy logic controller for a manipulator there are three techniques that may be applied: set the universal values for the largest possible case, set the universal values small to give good resolution and reduced steady state oscillations, and set the universal values using a combination of the first two. In the single 5x5 control matrix used above, a combination of the first two designs was used with an emphasis on reducing steady state oscillation. It was hypothesized, however, that by using two control laws within the same controller the best parts of each control technique could be exploited without compromising either advantage.

If the first control law is tuned to recognize large errors and is given a larger range of possible torque outputs, the initial time response can be reduced. If this controller is kept active, however, the controller may not be able to distinguish a zero error from a small error. This lack of resolution may cause the controller to command a torque response that is too large for the actual situation. This exaggerated torque often ends in the arm slewing past the desired point. The controller then 'catches' the arm, but again slews it past the desired point. This continues to happen resulting in a steady state oscillation instead of a settled steady state value. To combat the oscillation problem, the controller must have a tighter range of error and torque. The single 5x5 controller does this, but the initial time response is slow with the 2 second lag. Combining these two designs solves both problems.

The design of this structure gives control to the large control law for two seconds. After the initial slew has occurred, control passes to the small controller in order to damp out the system. The large matrix covers the ranges between  $\pm 0.5$  radians for angle error,  $\pm 1.5$  radians/second for angle error velocity, and  $\pm 0.6$  volts for torque. The membership curve equations for angle error are shown in Table 3.4, and the control matrix in Table 3.5. The membership curve equations for angle error velocity and torque are the same as for the single 5x5 matrix. The small matrix is exactly the same as for the single 5x5 matrix.

This method works in speeding up the initial time response and does damp out the system, but there are trade-offs in how the system is set up. If the hand-off between the large and small controller is done at the wrong time, or the system switches back and forth between the two controllers through out the slew, the system response may not be smooth. This action causes the overall settling time to increase rather than decrease. In this system, if the large controller is left active over 2 seconds the hand-off to the small

Table 3.6 Angle Error Large Five by Five Membership Curve Equations

Membership Curve Equations	
POS_MEDIUM	$\text{SIN}(\text{Pi}^*(\text{X}-0.5))^5$
POS_SMALL	$\text{SIN}(\text{Pi}^*(\text{X}-0.22))^5$
ZERO	$\text{SIN}(2*\text{Pi}^*(\text{X}+4.75))^5$
NEG_SMALL	$\text{SIN}(\text{Pi}^*(\text{X}+0.25))^5$
NEG_MEDIUM	$\text{SIN}(\text{Pi}^*(\text{X}+0.5))^5$

Table 3.7 Large Five by Five Controller Rule Matrix

	NM	NS	ZR	PS	PM
NM		NM	NM	PS	
NS		NS	NS	PM	
ZR		NM	ZR	PM	
PS		NM	PS	NS	
PM		NS	PM	PM	

controller is not smooth. The large controller is used to begin the initial slew, but if it is allowed to continue, the response of the system may require a large voltage. If the small controller is made active at this point, the manipulator will not respond as desired. Also, if the controllers are switched back and forth, the discontinuities in the voltage outputs may increase the oscillations of the system. The results of the hardware tests of this system are shown in Figure 3.21 and Figure 3.22. The time response of this controller is better than the single 5x5 matrix. The system's settling time decreases from 14 seconds to 13 seconds while the steady state error only increases from 3.1% to 3.7%. Due to the increased voltage input for the initial slew, the overshoot using this system rose from 52% to 62%. Surprisingly, however, the rise time remained constant at 1.14 seconds.

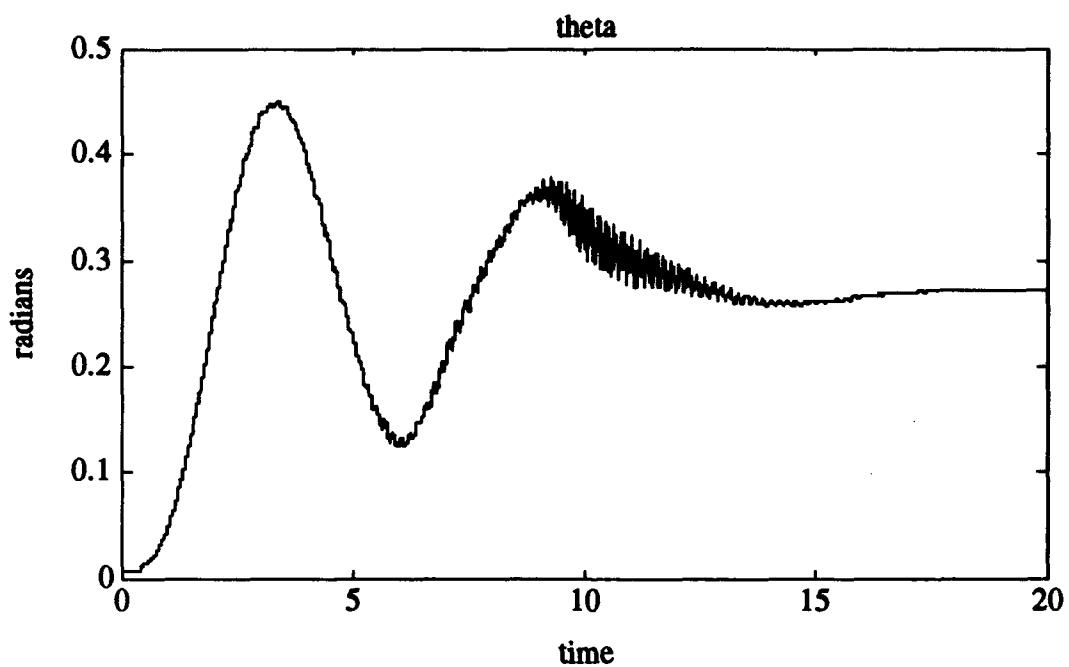


Figure 3.21 Theta Response for Double Control Law in Hardware, 15 Degree Slew

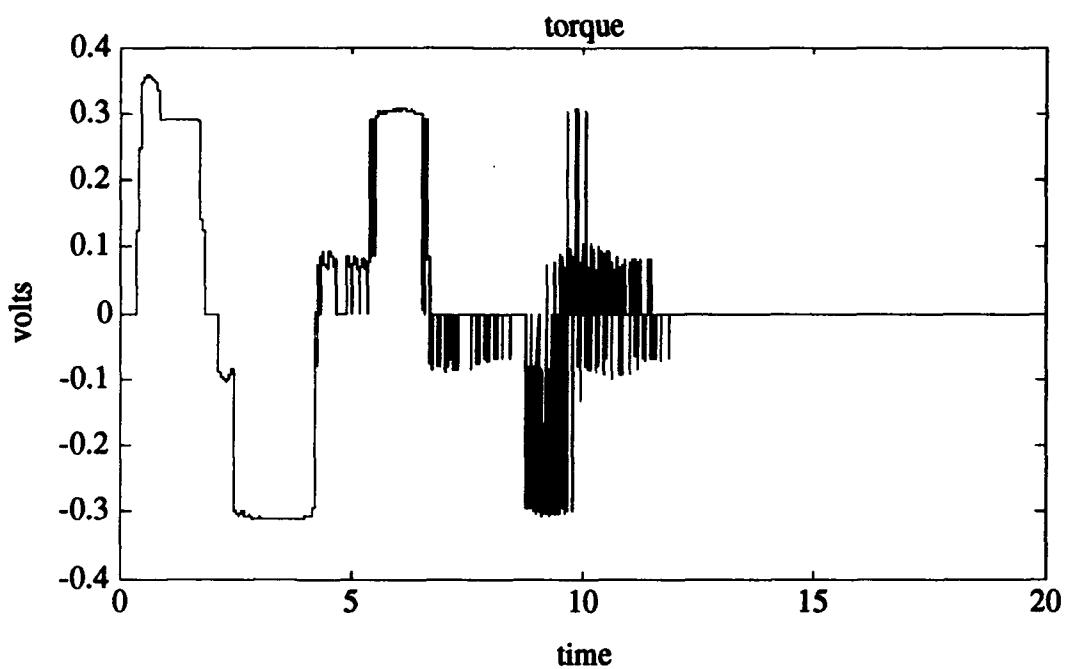


Figure 3.22 Torque Response for Double Control Law in Hardware, 15 Degree Slew

### 3.6.3 Robustness

To examine the performance robustness of the fuzzy logic controller, three off-design tests were accomplished. The first off-design test added a 1,160 gram mass to the end of the on-design manipulator, the second added a 830 gram mass to the end of the manipulator, and the third off-design test used a 1/16 inch aluminum arm instead of the on-design 1/8 inch aluminum arm. Both the PID controller and the two control law fuzzy logic controller were tested in both of the situations for a command slew of 15 degrees.

Figure 3.23 shows the theta response for the fuzzy logic controller to the first robustness test. As a matter of practicality in data acquisition, only the first 30 seconds of the test slews were recorded. However, it is obvious in this time that although the controller does not settle the system within the first 30 seconds, the system is indeed damped. Figure 3.25 shows the theta response for the PID controller for the same test. By noticing that the second overshoot is larger than the first, it is apparent that, unlike the fuzzy logic controller, the PID controller is not stable for this off-design condition.

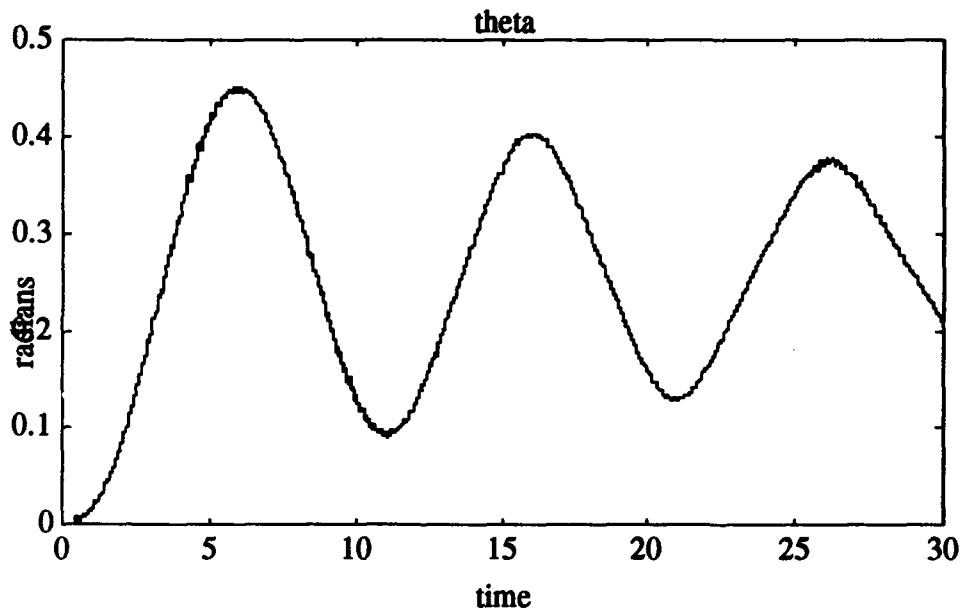


Figure 3.23 Theta Response for Fuzzy Logic Robustness Test I, 15 Degree Slew

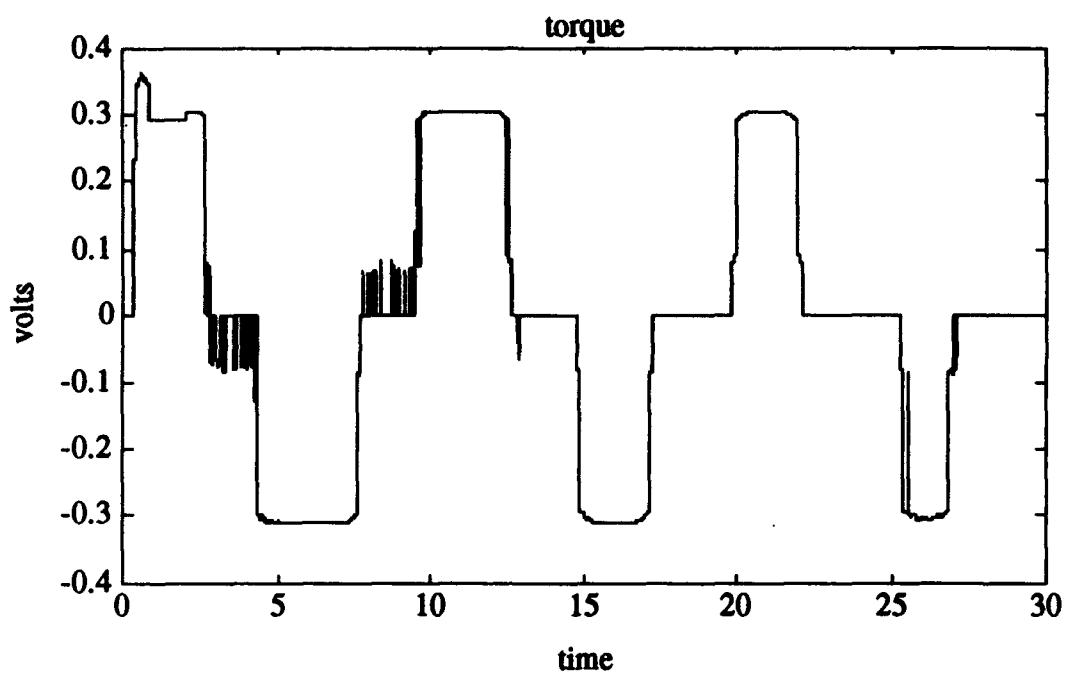


Figure 3.24 Torque Response for Fuzzy Logic Robustness Test I, 15 Degree Slew

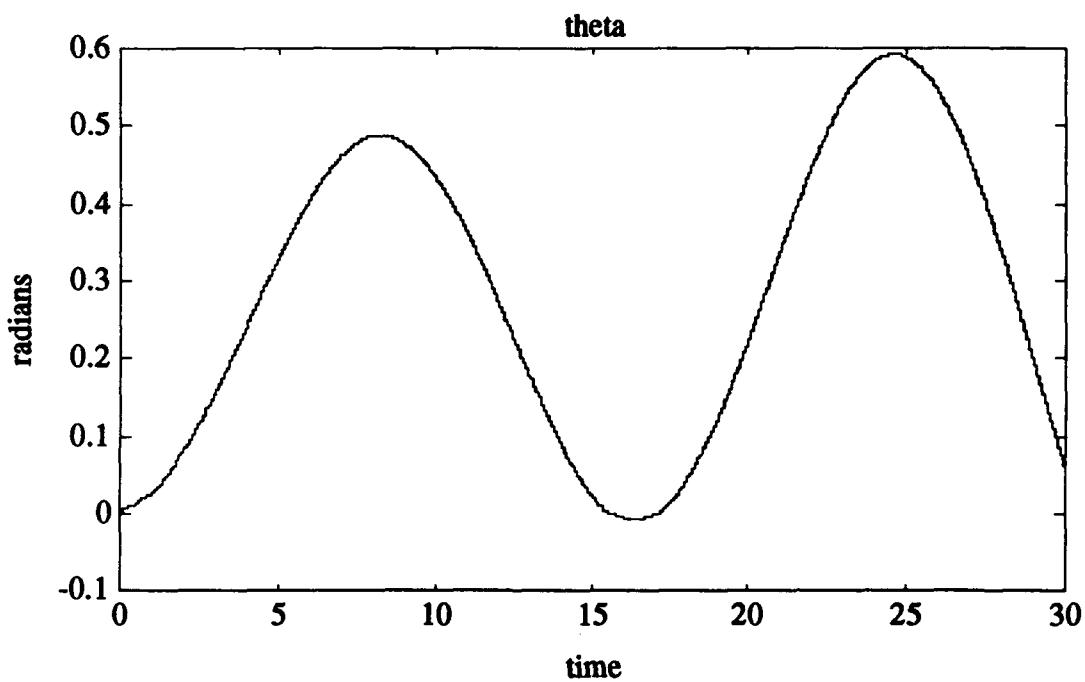


Figure 3.25 Theta Response for PID Robustness Test I, 15 Degree Slew

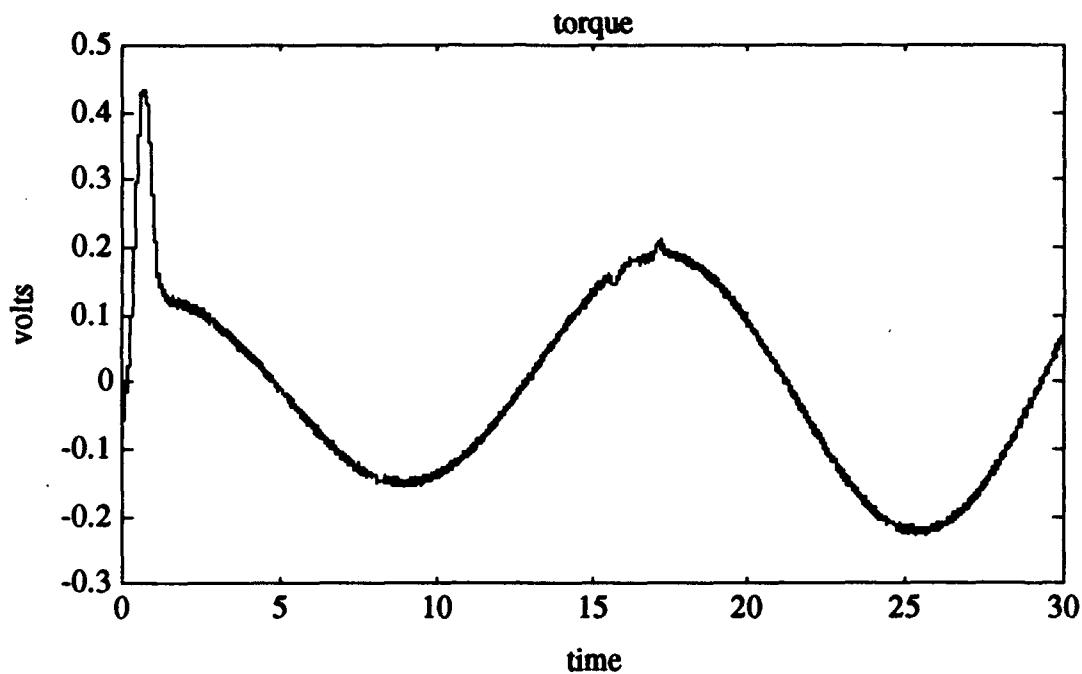


Figure 3.26 Torque Response for PID Robustness Test I, 15 Degree Slew

The theta response of the fuzzy logic controller for the second test is shown in Figure 3.27. Again the response is only shown for 30 seconds, but it is apparent that the system is stable. It is also apparent from the last undershoot that the system is settling into approximately 15 degrees. The theta response for the PID controller (Figure 3.29), shows that for this off-design case, the PID control law is able to stabilize the system. The PID controller appears to be taking longer to settle the system than the fuzzy logic controller, but overall the two controllers seem comparable for this off-design condition.

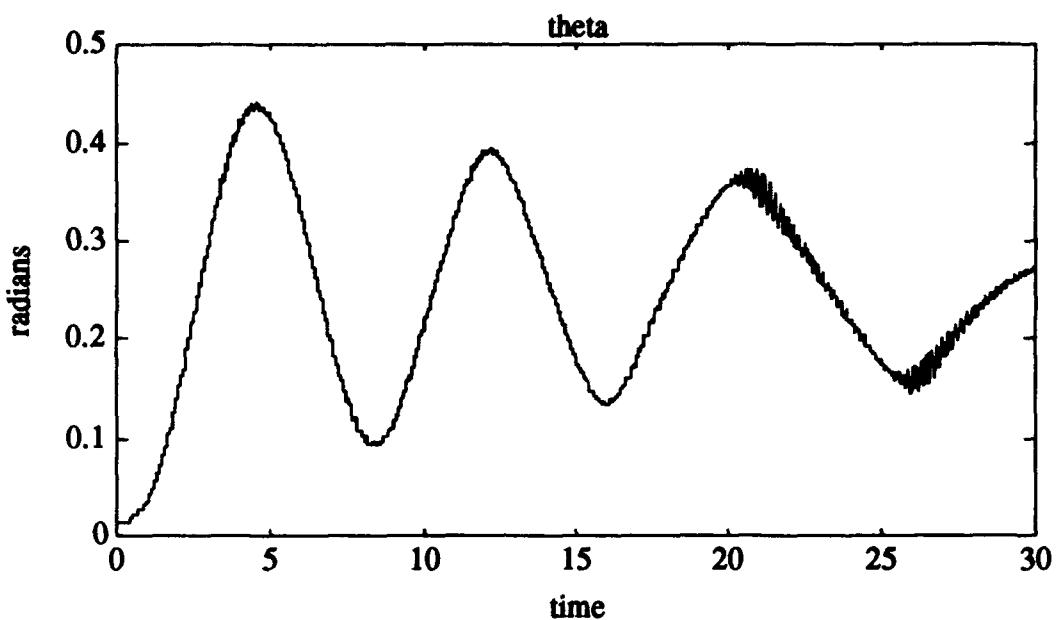


Figure 3.27 Theta Response for Fuzzy Logic Robustness Test II, 15 Degree Slew

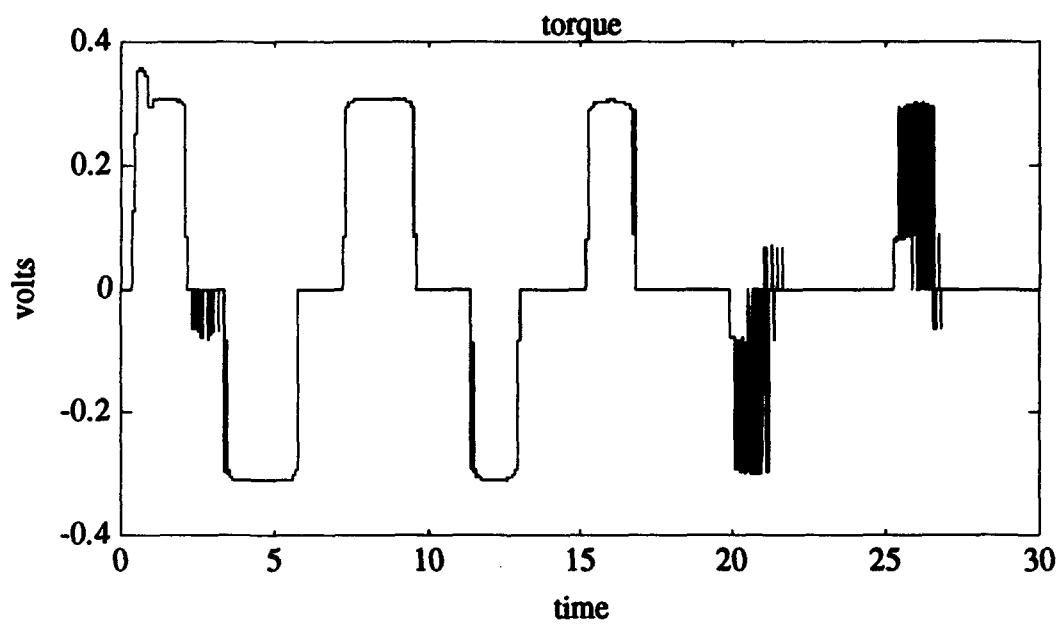


Figure 3.28 Torque Response for Fuzzy Logic Robustness Test II, 15 Degree Slew

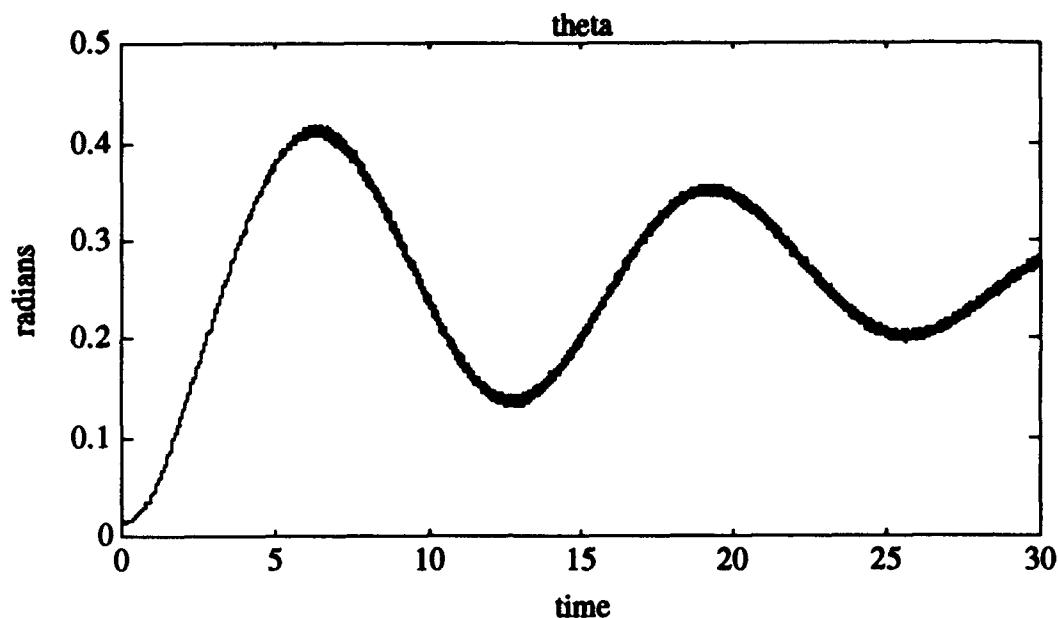


Figure 3.29 Theta Response for PID Robustness Test II, 15 Degree Slew

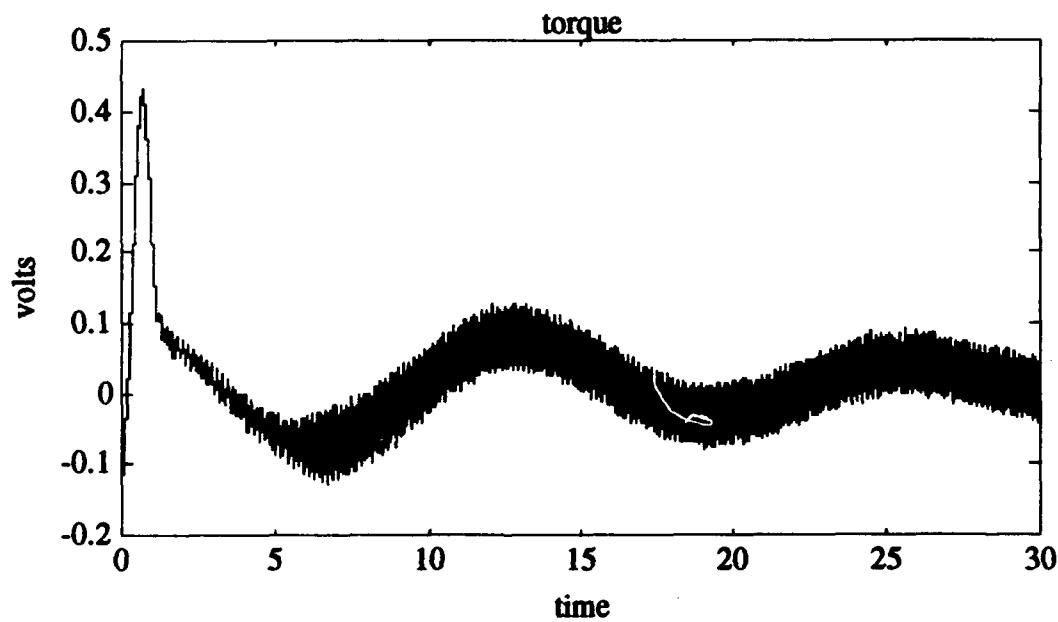


Figure 3.30 Torque Response for PID Robustness Test II, 15 Degree Slew

The fuzzy logic control law results for the third robustness test are shown in Figures 3.27 and 3.28. Figure 3.27 shows the theta results for the fuzzy logic controller. For this off-design case the controller stimulates the natural frequencies of the arm and

the overall performance is lower than the on-design condition, but the flexible arm damps out relatively quickly. The system settles to within 5% of the final value in 10 seconds, which is better than the on-design case, but the final value of .2415 radians is 7.75% lower than the desired value of .2618 radians. Like the steady state error, the rise time and percent overshoot of the system both increased versus the on-design case. The rise time increased almost 0.6 seconds to 1.72 seconds and the percent overshoot increased from 62 percent to 77 percent.

Figures 3.29 and 3.30 show the PID controller results for the third robustness test. For this configuration the PID controller works well overall with the exception that the flexible arm never actually damps out completely. The manipulator settles to within 5 percent of the reference value in 8 seconds, which is only 0.5 seconds slower than the on-design condition, but the steady state response has a  $\pm 1.0$  percent steady state error compared to a 2.7 percent constant error for the on-design case. Although the response is actually more accurate, the steady state oscillation shows the one weakness of the PID controller in this test. The rise time of this system is 0.85 seconds faster than the original value, but the overshoot increases from 5.71 percent to 15 percent.

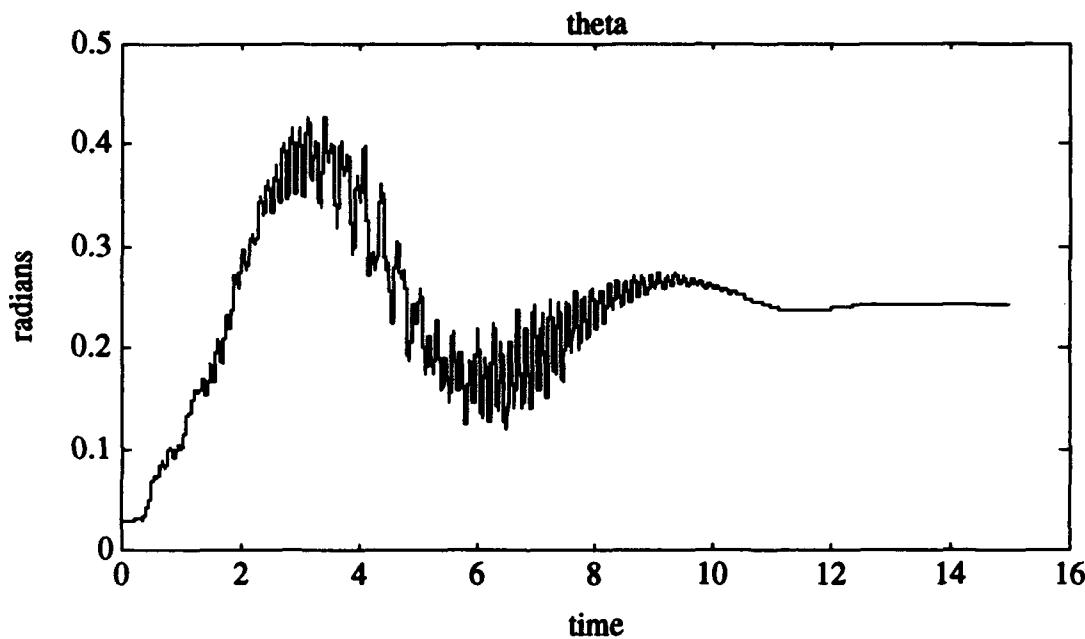


Figure 3.31 Theta Response for Fuzzy Logic Robustness Test III, 15 Degree Slew

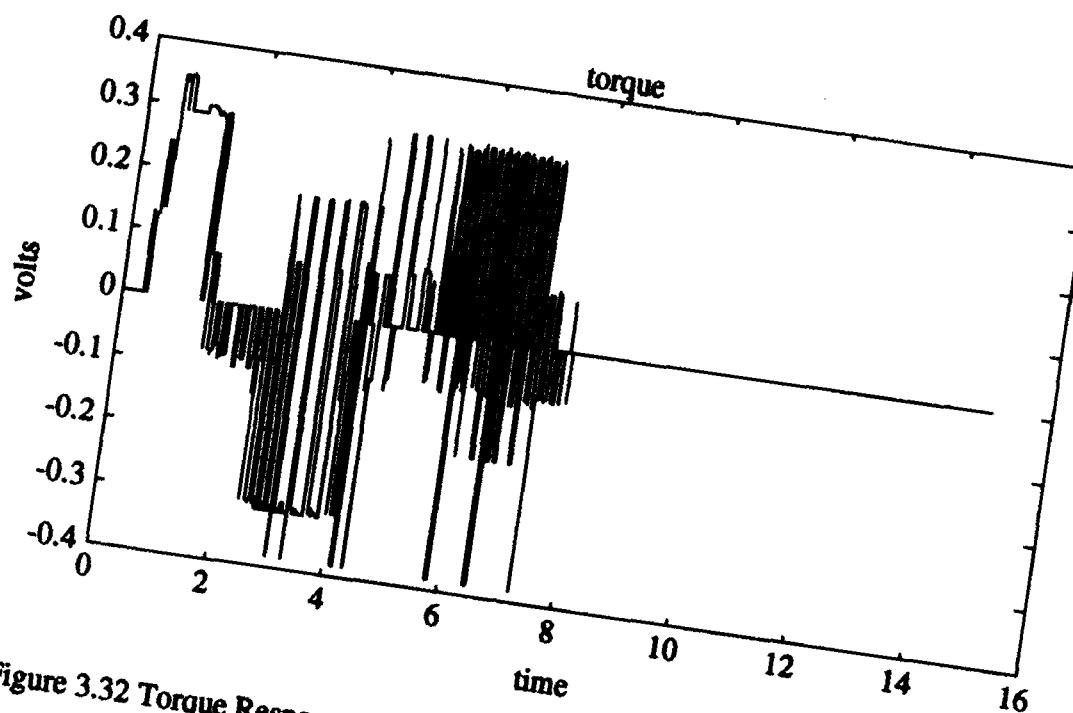


Figure 3.32 Torque Response for Fuzzy Logic Robustness Test III, 15 Degree Slew

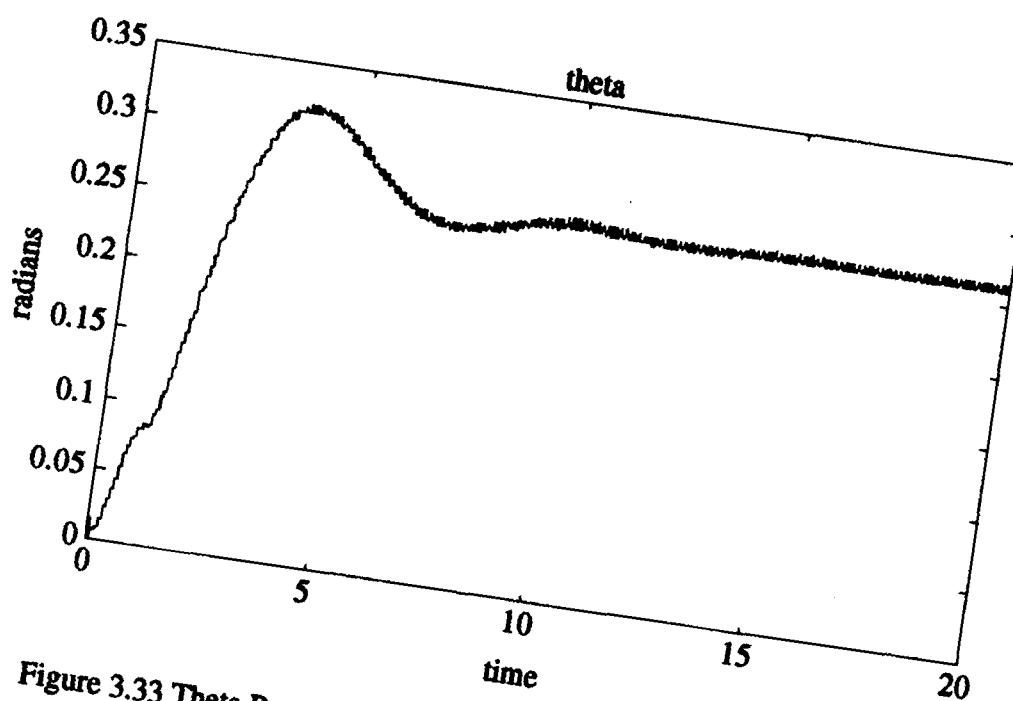


Figure 3.33 Theta Response for PID Robustness Test III, 15 Degree Slew

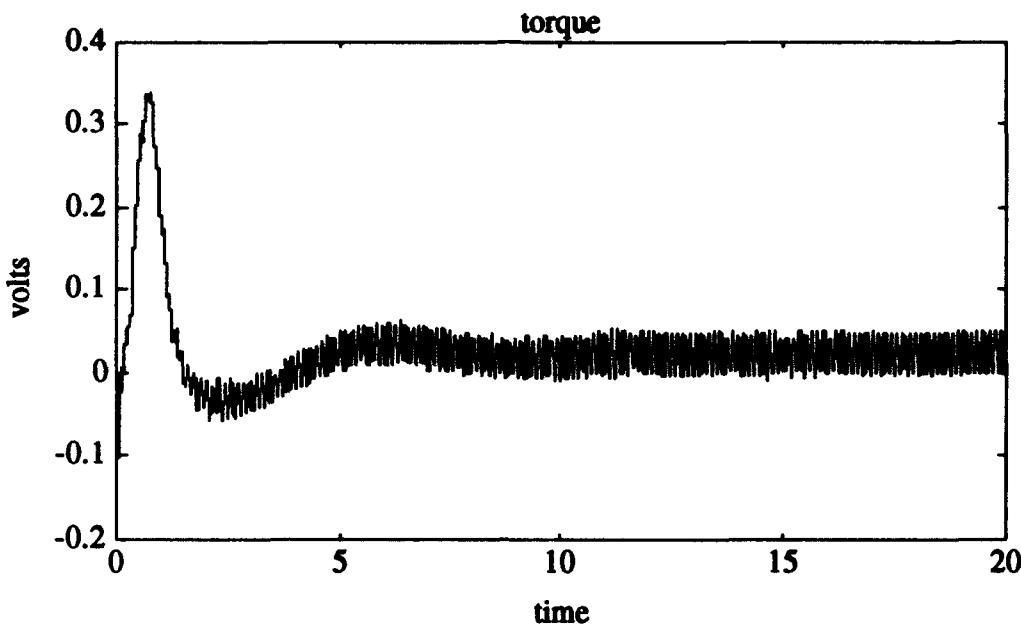


Figure 3.34 Torque Response for PID Robustness Test III, 15 Degree Slew

### 3.6.4 Seven by Seven Control Matrices

As described for the 5x5 matrices, one method to reduce steady state error was to make the membership curves around zero smaller to give better resolution for a true zero error. It follows that if many small membership curves are used instead of a few large membership curves, the resolution for the entire universe of values will improve. If this approach is used for not only angle error, but also angle error velocity and torque the time response of the system will improve instead of just the steady state error. In this section the results of a control law using a single 7x7 control matrix and a control law using two 7x7 control matrices will be discussed. Unfortunately, due to computer software problems in downloading these control laws to the real-time control computer, only computer simulations of these control laws were possible.

The 7x7 matrices use the same five divisions used in the 5x5 matrices plus positive large and negative large. For the single 7x7 controller, the seven divisions are spread from  $\pm 0.5$  radians for angle error,  $\pm 2$  radians/second for angle error velocity, and  $\pm 0.3$  volts for torque. Because of the added number of membership curves, all of the membership curves have the same shape for all three variables. The equations for the

curves for this controller are shown in Table 3.8 while the rule matrix is shown in Table 3.9.

Table 3.8 Single Seven by Seven Controller Membership Curves

Membership Curve Equations	
POS_LARGE	$\text{SIN}(\text{Pi}^*(\text{X}-0.5))^9$
POS_MEDIUM	$\text{SIN}(\text{Pi}^*(\text{X}-0.5))^5$
POS_SMALL	$\text{SIN}(\text{Pi}^*(\text{X}-0.25))^5$
ZERO	$\text{SIN}(\text{Pi}^*\text{X})^3$
NEG_SMALL	$\text{SIN}(\text{Pi}^*(\text{X}+0.25))^5$
NEG_MEDIUM	$\text{SIN}(\text{Pi}^*(\text{X}+0.5))^5$
NEG_LARGE	$\text{SIN}(\text{Pi}^*(\text{X}+0.5))^9$

Table 3.9 Single Seven by Seven Controller Rule Matrix

	NL	NM	NS	ZR	PS	PM	PL
NL				NM			
NM			NM		NL		
NS			NS	NL	PL		
ZR		NL	NM	ZR	PM	PL	
PS			NL	PL	PS		
PM			PL		PM		
PL				PM			

The simulation results for the single 7x7 controller are shown in Figure 3.29 and Figure 3.30. In general, when comparing the results of the hardware tests using the single 5x5 controller to the results of the simulation tests using the single 7x7 controller, the performance values improve for the larger matrix. Although the steady state error increased from 3.1 percent to 3.7 percent, a difference of only 0.1 degrees, the rise time decreased by 3.5 seconds to 10.5 seconds and the percent overshoot decreased from 52 percent to only 40 percent. The rise time actually rose 0.18 seconds, but due to the initial 2 second lag for the 5x5 controller response, the system reached 90 percent of the steady state value faster overall with the 7x7 system.

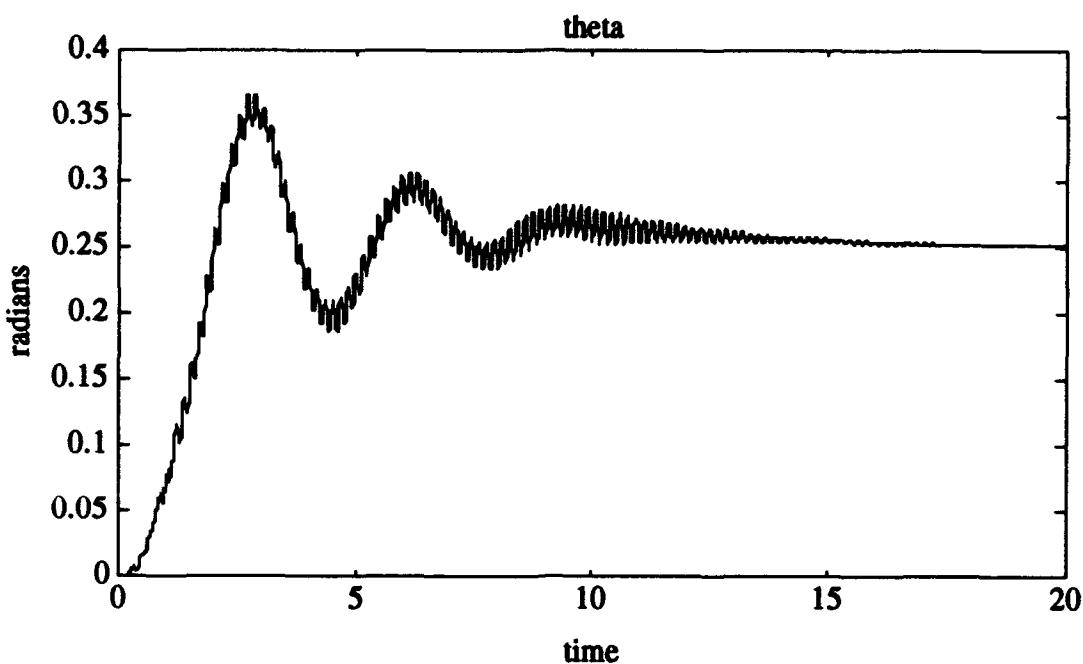


Figure 3.35 Seven by Seven Theta in Simulation, 15 Degree Slew

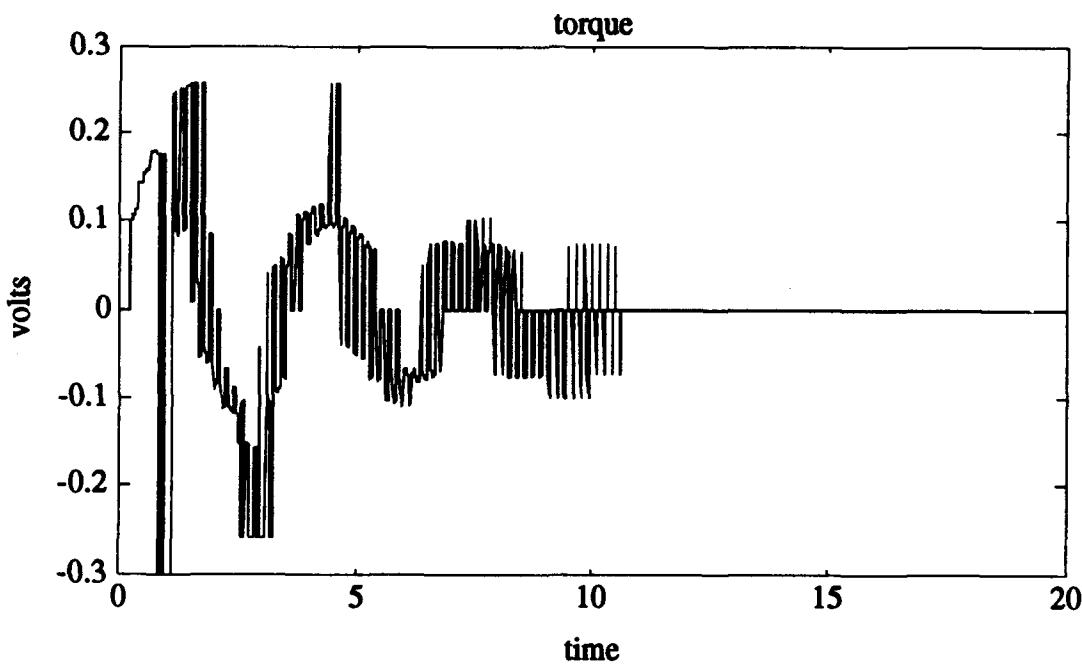


Figure 3.36 Seven by Seven Torque in Simulation, 15 Degree Slew

The results of the single 7x7 matrix control law were then compared to a double 7x7 matrix control law. The rule matrix, membership curves, and universal values for the

large controller are the same as the single 7x7 controller with the exception that the universal range for the angle error is  $\pm 0.5$  radians instead of  $\pm 0.3$  radians. The small controller uses the same membership curves, but the rule matrix and universal values are changed. The universal values of angle error range from  $\pm 0.17$  radians, the values of angle error velocity range from  $\pm 0.45$  radians/second, and the values of torque range from  $\pm 0.1$  volts. The rule matrix for the small controller is shown in Table 3.10.

Table 3.10 Small Seven by Seven Controller Rule Matrix

	NL	NM	NS	ZR	PS	PM	PL
NL				NM			
NM			NS		PM		
NS			NS	NM	PM		
ZR	NM		NM	ZR	PM		PM
PS			NM	PM	PS		
PM			NM		PS		
PL				PM			

The results of the double 7x7 control law are the best of any of the examined controllers. The theta and torque responses of the system (Figures 3.31 and 3.32) show the performance values of this controller. The steady state error is only 2.6 percent, the rise time is only 1 second, and the settling time is only 7 seconds. While these are the fastest responses of any of the controllers, there is still only a 42 percent overshoot.

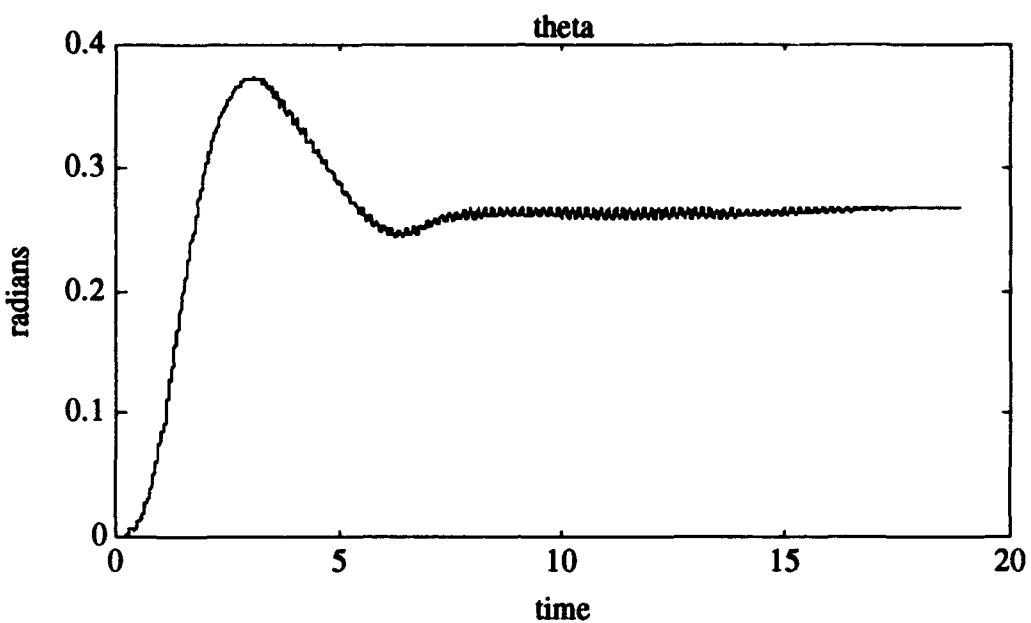


Figure 3.37 Double Seven by Seven Theta in Simulation, 15 Degree Slew

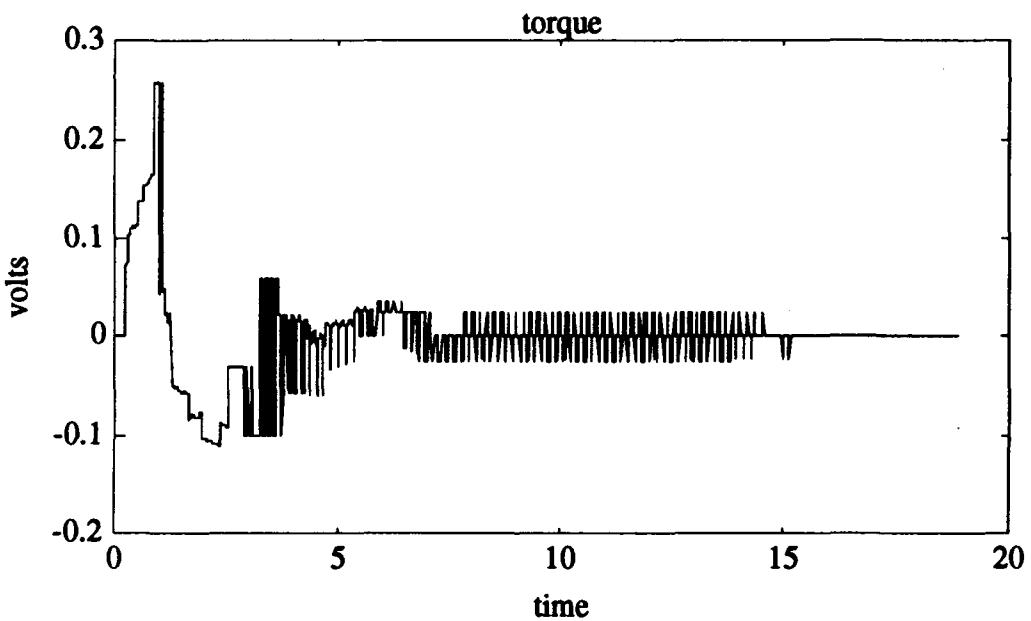


Figure 3.38 Double Seven by Seven Torque in Simulation, 15 Degree Slew

## Chapter 4: Conclusions and Recommendations

### 4.1 Conclusions

The purpose of this work was to explore different fuzzy logic controller configurations and implement them on a single link, flexible robot arm. The first step in this process was to validate the plant model and the AC100-C30 system by implementing a PID controller in simulation as well as hardware. When this was accomplished, four fuzzy logic controllers were designed. The performance values of all of the tested controllers are shown in Table 4.1.

The first controller used a single 5x5 rule matrix while the second controller used two 5x5 rule matrices. Using the two rule matrices in the second controller allowed the system to respond faster to the initial slew command than the single rule matrix controller while still settling to a final value. To help reduce steady state error while using the 5x5 rule matrices, the membership curves corresponding to zero error were reduced in width in comparison to the rest of the membership curves.

The third and fourth controllers used 7x7 rule matrices. Due to the increased granularity added by the smaller membership curves associated with the larger matrices, the shapes of the individual membership curves could remain constant while still achieving good steady state error performance. As with the 5x5 rule matrix controllers, the first 7x7 controller used a single rule matrix while the second control law used two 7x7 controllers. Unfortunately, due to a software error between simulation and hardware-in-the-loop implementation, the 7x7 controllers could not be implemented in hardware. The performance values of these two controllers in simulation, however, showed the advantage of a larger control matrix.

The robustness of the PID controller as well as the double 5x5 control law fuzzy logic controller were examined in three tests. The first test was a 15 degree slew with a 1160 gram mass at the tip of the manipulator. The second test was a 15 degree slew with a 830 gram mass at the tip of the manipulator. The third test was a 15 degree slew with a 1/16th inch thick manipulator instead of the on-design 1/8th inch thick manipulator. The results of these tests, listed in Table 4.1, show that the 5x5 rule matrix fuzzy logic controller is indeed stable for all of these conditions while the PID controller actually

goes unstable in the first test. Although the fuzzy logic controller does not settle the manipulator within 30 seconds in two of these tests, and has a 7 percent steady state error in the third, it must be remembered that the simulation results show the performance of the 7x7 rule matrix control laws superior to the 5x5 rule matrix. It is probable that if these control laws were implemented in the hardware that they would perform better than the 5x5 control laws. Overall, however, the fuzzy logic controllers were able to show the robust characteristics inherent to the control law.

Table 4.1 Performance Response

	Steady State Error	Rise Time (Seconds)	Overshoot	Settling Time (Seconds)
PID	2.7%	2.81	5.71%	7.5
5x5	3.1%	1.142	52%	14
2x5x5	3.7%	1.14	62%	13
7x7	3.7%	1.32	40%	10.5
2x7x7	2.6%	1	42%	7
Robust I	PID Unstable Fuzzy Logic Stable			
Robust II	Both Stable			
PID Robust III	1%	1.96	15%	8
Fuzzy Robust III	7.75%	1.72	77%	10

#### 4.2 Recommendations

There are a couple different directions for future work for fuzzy logic and the flexible manipulator. The first possibility would be to continue working on the single link arm. Although this research explored the basic concepts of fuzzy logic and demonstrated the general performance and robustness of fuzzy logic, it did not explore the complexities of fuzzy logic. Some areas that could be used in the future are adding weights to the rules and using qualifiers for membership curves. By using these aspects of fuzzy logic, the 5x5 rule matrix controllers may be improved in both on-design and off-design

performance. Along with adding these areas in the control law, further work should be done with Integrated Systems Inc. to allow 7x7 fuzzy logic rule matrices to be used with the hardware-in-the-loop capabilities of the AC100-C30 system. If ISI is not able to debug their C code generator it may still be profitable for C code to be hand written to accommodate the 7x7 controller.

A second direction for future work is the addition of a second link to the manipulator. At this time, a PID controller is operational in the University of Washington Control System Laboratory robot arm test facility for a two link flexible arm. The next step could be to implement a fuzzy logic controller on this hardware and compare the performance capabilities of the controllers.

Finally, a third direction for future work concerns the sensors in the hardware. If the manipulator is to be slewed and damped as quickly as possible, real-time knowledge of the manipulators dynamics would be helpful. For this research, only the base angle, theta, was known. If, as in the work at the University of Akron [4], tip displacement was known as well as base angle, the controller could actively damp some of the flexible modes as well as slew the manipulator to the desired angle.

## BIBLIOGRAPHY

- [1] Evers S. *The UWCSL Two-Link Manipulator Facility: System Description*. University of Washington Controls System Laboratory Report, 1993.
- [2] Gupta, A.R. *Controller Design Using Fuzzy Logic (RT/Fuzzy)*. Integrated Systems, Inc.,
- [3] Integrated Systems, Inc. *Real-time Fuzzy Logic Block User's Guide*, 1991.
- [4] Lin, Y.J. and T.S. Lee. *Modeling for Fuzzy Logic Control of Deformable Manipulators*. Proceedings of the American Controls Conference, San Fransisco, Ca. June 1993.
- [5] Oakley, C.M. *Experiments in Modelling and End-Point Control of Two-Link Flexible Manipulators*. Ph.D. Dissertation, Mechanical Engineering Dept., Stanford University, April 1991.
- [6] Vagners, J. *Control Systems Laboratory Development*. Proceedings of the 1993 American Controls Conference, San Francisco, Ca. June 1993.

## Appendix A: Plant Generation Programs

### I. Parameters

```
// Description: Returns SLA inertial and geometric parameters for system
// configuration A. This configuration is a long single flexible
// link with no tip mass loading.
//
//
// Input Variables
// n = number of flexible elements for link
//
// Output Variables
// Ih1 = Inertia of shoulder hub about its mass center
// Ih2 = Inertia of tip hub about its mass center
// Ili = Inertia of ith link about its mass center
// It1 = Inertia of tip (bracket and mass) about its mass center
// mh1 = mass of shoulder hub
// mh2 = mass of tip hub
// mli = mass of ith link
// mt1 = mass of tip (bracket and pad)
// ch1 = distance from shoulder shaft to shoulder hub center of mass
// ct1 = distance from tip hub centroid to tip center of mass
// oh1 = distance from shoulder shaft to shoulder hub attach point of link 1
// ot1 = distance from tip hub centroid to tip attach point
// li = length of each rigid segment of link i
// ki = torsional spring stiffness between ith link segments
// bi = torsional viscous damping between ith link segments

// Revision: 20JULY93
// by Steve Evers
n1=20
th1_nom=45*pi/180

// shoulder hub parameters
Ih1 = 4.1238e-03; //kg*m**2
mh1 = 1.1618; //kg
ch1 = 5.5959e-05; //m
oh1 = 5.8420e-02; //m

// link 1 parameters
// long flexible link, no mass intensifiers
mli = 415.5/1000; //kg (measured)
ml1 = mli/n1; //kg
l1 = ((40.5-1-1.5)/12)*0.3048; //m (measured)
l1 = l1/n1; //m
Il1 = (ml1*l1**2)/12; //kg**2

// tip 1 parameters
// tip is composed of bracket, pad, and link grip length
// tip location is defined as centerline of bracket base bolts
mt1 = 2.9842e-01; //kg
ct1 = 4.3807e-04; //m
ot1 = 6.3500e-03; //m
```

```
ct1 = 4.3807e-04; //m
ot1 = 6.3500e-03; //m
It1 = 1.1045e-03; //kg*m**2

// no motor hub2 loaded
// all end point loading is associated with tip parameters
lh2 = 0;
mh2 = 0;

// beam lateral stiffness properties
// AL 6061 stiffness
E = 70e9; //N/m**2 = Pa
h = (1.5/12)*0.305; //m
b = ((1/8)/12)*0.305; //m
I = h*b**3/12; //m**4
Ell1 = 7.9;
k1 = 3*Ell1*(n1*(n1-1)*(2*n1-1)/6)/(n1**2*n1*l1);

// beam structural damping
// tuned to 40th order plant model Stanford experimental results
b = 0.01;
b1 = 300*b/n1;
```

## II. Plant Generator

```

// Description: Synthesizes a lumped spring/mass/damper (LSMD) state space
// model for the UW flex-flex SLA in the following form:
//
//  $\frac{d}{dt}\{x\} = A\{x\} + B\{u\}$ 
//  $\{del_y\} = C\{x\} + D\{u\}$ 
//
// Note: both  $\{x\}$  and  $\{del_y\}$  are referenced to the LOCAL
// coordinate system determined by the state vector:
//  $x = \{0 \dots 0 \, th1\_nom \, 0 \dots 0\}'$ 
//
//  $x = \{dth1 \, dq11 \dots dq1(n1-1) \, th1 \dots q1(n1-1)\}'$ 
//  $\{del_y\} = \{x \, del_xep \, del_yep\}'$ 
//  $\{u\} = \{T1\}'$ 
//  $T1$  = shoulder torque
//  $th1$  = shoulder joint base angle
//  $del_xep$  = x end point perturbed dynamics
//  $del_yep$  = y end point perturbed dynamics
// ALL UNITS IN METRIC
//
// Input Variables:  $n1$  = number of flexible elements for first link
//  $th1\_nom$  = nominal shoulder joint angle to linearize about
//
// Output Variables:  $[A, B, C, D]$  = state space realization in locally prescribed
// coordinate system.

// Revision: 28JUNE93
// by Steve Evers

// load inertia/spring/mass parameters
SLA_plant_param_f

// mass matrices
for i = 2:n1-1, ...
  M_inertias11(i,i) = I11; ...
end
M_inertias11(1,1) = Ih1+I11;
M_inertias11(n1,n1) = I11+It1+Ih2;
M_inertias = [M_inertias11];

for i = 2:n1-1, ...
  M_hubs_diag11(i,i) = mh2*I11**2; ...
end
M_hubs_diag11(1,1) = mh1*ch1**2+mh2*(oh1+I1)**2;
M_hubs_diag11(n1,n1) = mh2*(I1+ot1)**2;
M_hubs_diag = [M_hubs_diag11];

for i = 2:n1, ...
  M_link1_diag11(i,i) = (1/4)*m11*I11**2+(n1-i)*m11*I11**2; ...
end
// Stanford element I believe to be incorrect

```

```

//M_link1_diag11(1,1) = ml1*(oh1+(l1/2))**2+n1*ml1*(oh1+l1)**2;
// My element
M_link1_diag11(1,1) = ml1*(oh1+(l1/2))**2+(n1-1)*ml1*(oh1+l1)**2;
M_link1_diag = [M_link1_diag11];

M_tip1_diag = 0*ones(n1);
for i = 2:n1-1, ...
    M_tip1_diag(i,i) = mt1*l1**2, ...
end
M_tip1_diag(1,1) = mt1*(oh1+l1)**2;
M_tip1_diag(n1,n1) = mt1*(l1+ot1+ct1)**2;

M_hubs_off_diag11 = 0*ones(n1,n1);
for i = 2:n1-2, ...
    for j = i+1:n1-1, ...
        M_hubs_off_diag11(i,j) = mh2*l1**2, ...
    end
end
for i = 2:n1-1, ...
    M_hubs_off_diag11(1,i) = mh2*(oh1+l1)*l1, ...
    M_hubs_off_diag11(i,n1) = mh2*l1*(l1+ot1), ...
end
M_hubs_off_diag11(1,n1) = mh2*(oh1+l1)*(l1+ot1);
M_hubs_off_diag11 = M_hubs_off_diag11+M_hubs_off_diag11';
M_hubs_off_diag = [M_hubs_off_diag11];

M_link1_off_diag11 = 0*ones(n1,n1);
for i = 2:n1-1, ...
    for j = i+1:n1, ...
        M_link1_off_diag11(i,j) = (1/2)*ml1*l1**2+(n1-j)*ml1*l1**2, ...
    end
end
for i = 2:n1, ...
    M_link1_off_diag11(1,i) = (1/2)*ml1*(oh1+l1)*l1+(n1-i)*ml1*(oh1+l1)*l1, ...
end
M_link1_off_diag11 = M_link1_off_diag11+M_link1_off_diag11';
M_link1_off_diag = [M_link1_off_diag11];

M_tip1_off_diag11 = 0*ones(n1,n1);
for i = 2:n1-2, ...
    for j = i+1:n1-1, ...
        M_tip1_off_diag11(i,j) = mt1*l1**2, ...
    end
end
for i = 2:n1-1, ...
    M_tip1_off_diag11(1,i) = mt1*(oh1+l1)*l1, ...
    M_tip1_off_diag11(i,n1) = mt1*l1*(l1+ot1+ct1), ...
end
M_tip1_off_diag11(1,n1) = mt1*(oh1+l1)*(l1+ot1+ct1);
M_tip1_off_diag11 = M_tip1_off_diag11+M_tip1_off_diag11';
M_tip1_off_diag = [M_tip1_off_diag11];

```

```

M =
M_inertias+M_hubs_diag+M_link1_diag+M_tip1_diag+M_hubs_off_diag+M_link1_off
_diag+M_tip1_off_diag;

// state transformation of mass matrix
for i = 1:n1, ...
  for j = 1:n1, ...
    Mbar(i,j) = sum(M(i,j:n1)); ...
  end
end

//nargin=3
//if nargin >= 5, ...
// k1 = k1*kf; ...
//end

// stiffness matrix
for i = 1:(n1-1), ...
  KK1(i,i+1) = -k1; ...
  KK1(i+1,i+1) = k1; ...
end
K = [KK1];

// damping matrix
for i = 1:(n1-1), ...
  D1(i,i+1) = -b1; ...
  D1(i+1,i+1) = b1; ...
end
D = [D1];

// input torque matrix
T = 0*ones(n1,1);
T(1,1) = 1;

// state space realization
// Mbar{d/dt(x)} = A{x} + T{u}
// {y} = C{x} + D{u}
Mbar = [Mbar 0*ones(n1, ...
  0*ones(n1) eye(n1));
A = [-1*D -1*K;
  eye(n1) 0*ones(n1)];
T = [T;0*ones(n1,1)];
Abar = Mbar\A;
Bbar = Mbar\T;
C = [eye(2*n1);
  0*ones(2,2*n1)];
// Stanford model had error in the term relating to the 1st element of each link
// del_x component
C(2*n1+1,n1+1) = -(oh1+l1)*sin(th1_nom)-(n1-1)*l1*sin(th1_nom)-ot1*sin(th1_nom);
cc = -(n1-1)*l1*sin(th1_nom)-ot1*sin(th1_nom);
for i = 1:(n1-1), ...
  C(2*n1+1,n1+i+1) = cc-(1-i)*l1*sin(th1_nom); ...

```

```
end
// del_y component
C(2*n1+2,n1+1) = (oh1+l1)*cos(th1_nom)+(n1-1)*l1*cos(th1_nom)+ot1*cos(th1_nom);
cc = (n1-1)*l1*cos(th1_nom)+ot1*cos(th1_nom);
for i = 1:(n1-1),...
    C(2*n1+2,n1+i+1) = cc+(1-i)*l1*cos(th1_nom);...
end
D = 0*ones(2*n1+2,1);
```

## Appendix B: TLA Model Development

This section, taken directly from the work by Evers, summarizes the steps taken in developing a 6 DOF dynamic model for the TLA manipulator. The SLA is then derived from this development. The model parameters used are those described in Figures 2.1 and 2.2. Three elements per link are used to develop the 6 DOF model.

Lagranges approach will be used in this derivation of system dynamic equation of motion. Using the model structure of Figure 2.2, with 3 (equal length and mass) elements per link, the mass center velocities of each element are as follows.

$$\begin{aligned}
 v_{h1}^2 &= c_{h1}^2 u_1^2 \\
 v_{l_{11}}^2 &= (o_{h1} + \frac{l1}{2})^2 u_1^2 \\
 v_{l_{12}}^2 &= u_1^2 (o_{h1} + l1)^2 + u_2^2 (\frac{l1}{2})^2 + u_1 u_2 (o_{h1} + l1) l1 \cos(-q_{11}) \\
 v_{l_{13}}^2 &= u_1^2 (o_{h1} + l1)^2 + u_2^2 l1^2 + u_3^2 (\frac{l1}{2})^2 \\
 &+ 2u_1 u_2 (o_{h1} + l1) l1 \cos(-q_{11}) + u_1 u_3 (o_{h1} + l1) l1 \cos(-q_{11} - q_{12}) \\
 &+ u_2 u_3 l1^2 \cos(-q_{12}) \\
 v_{l1}^2 &= u_1^2 (o_{h1} + l1)^2 + u_2^2 l1^2 + u_3^2 (l1 + o_{n1} + c_{n1})^2 + \\
 &+ 2u_1 u_2 (o_{h1} + l1) l1 \cos(-q_{11}) + 2u_1 u_3 (o_{h1} + l1) (l1 + o_{n1} + c_{n1}) \cos(-q_{11} - q_{12}) \\
 &+ 2u_2 u_3 l1 (l1 + o_{n1} + c_{n1}) \cos(-q_{12}) \\
 v_{h2}^2 &= u_1^2 (o_{h1} + l1)^2 + u_2^2 l1^2 + u_3^2 (l1 + o_{n1})^2 + u_4^2 c_{h2}^2 \\
 &+ 2u_1 u_2 (o_{h1} + l1) l1 \cos(-q_{11}) + 2u_1 u_3 (o_{h1} + l1) (l1 + o_{n1}) \cos(-q_{11} - q_{12}) \\
 &+ 2u_1 u_4 (o_{h1} + l1) c_{h2} \cos(-q_{11} - q_{12} - \theta_2) + 2u_2 u_3 l1 (l1 + o_{n1}) \cos(-q_{12}) \\
 &+ 2u_2 u_4 l1 c_{h2} \cos(q_{12} - \theta_2) + 2u_3 u_4 (l1 + o_{n1}) c_{h2} \cos(-\theta_2) \\
 v_{l_{21}}^2 &= u_1^2 (o_{h1} + l1)^2 + u_2^2 l1^2 + u_3^2 (l1 + o_{n1})^2 + u_4^2 (o_{h2} + \frac{l2}{2})^2 \\
 &+ 2u_1 u_2 (o_{h1} + l1) l1 \cos(-q_{11}) + 2u_1 u_3 (o_{h1} + l1) (l1 + o_{n1}) \cos(-q_{11} - q_{12}) \\
 &+ 2u_1 u_4 (o_{h1} + l1) (o_{h2} + \frac{l2}{2}) \cos(-q_{11} - q_{12} - \theta_2) + 2u_2 u_3 l1 (l1 + o_{n1}) \cos(-q_{12}) \\
 &+ 2u_2 u_4 l1 (o_{h2} + \frac{l2}{2}) \cos(-q_{12} - \theta_2) + 2u_3 u_4 (l1 + o_{n1}) (o_{h2} + \frac{l2}{2}) \cos(-\theta_2) \\
 v_{l_{22}}^2 &= u_1^2 (o_{h1} + l1)^2 + u_2^2 l1^2 + u_3^2 (l1 + o_{n1})^2 + u_4^2 (o_{h2} + l2)^2 + u_5^2 (\frac{l2}{2})^2 \\
 &+ 2u_1 u_2 (o_{h1} + l1) l1 \cos(-q_{11}) + 2u_1 u_3 (o_{h1} + l1) (l1 + o_{n1}) \cos(-q_{11} - q_{12}) \\
 &+ 2u_1 u_4 (o_{h1} + l1) (o_{h2} + l2) \cos(-q_{11} - q_{12} - \theta_2) \\
 &+ u_1 u_5 (o_{h1} + l1) l2 \cos(-q_{11} - q_{12} - \theta_2 - q_{21}) \\
 &+ 2u_2 u_3 l1 (l1 + o_{n1}) \cos(-q_{12}) + 2u_2 u_4 l1 (o_{h2} + l2) \cos(-q_{12} - \theta_2) \\
 &+ u_2 u_5 l1 l2 \cos(-q_{12} - \theta_2 - q_{21}) + 2u_3 u_4 (l1 + o_{n1}) (o_{h2} + l2) \cos(-\theta_2) \\
 &+ u_3 u_5 (l1 + o_{n1}) l2 \cos(-\theta_2 - q_{21}) + u_4 u_5 (o_{h2} + l2) l2 \cos(-q_{21})
 \end{aligned}$$

$$\begin{aligned}
v_{l_3}^2 &= u_1^2(o_{h1} + l1)^2 + u_2^2l1^2 + u_3^2(l1 + o_{n1})^2 + u_4^2(o_{h2} + l2)^2 + u_5^2l2^2 + u_6^2\left(\frac{l2}{2}\right)^2 \\
&+ 2u_1u_2(o_{h1} + l1)l1\cos(-q_{11}) + 2u_1u_3(o_{h1} + l1)(l1 + o_{n1})\cos(-q_{11} - q_{12}) \\
&+ 2u_1u_4(o_{h1} + l1)(o_{h2} + l2)\cos(-q_{11} - q_{12} - \theta_2) + 2u_1u_5(o_{h1} + l1)l2\cos(-q_{11} - q_{12} - \theta - q_{21}) \\
&+ u_1u_6(o_{h1} + l1)l2\cos(-q_{11} - q_{12} - \theta - q_{21} - q_{22}) + 2u_2u_3l1(l1 + o_{n1})\cos(-q_{12}) \\
&+ 2u_2u_4l1(o_{h2} + l2)\cos(-q_{12} - \theta_2) + 2u_2u_5l1l2\cos(-q_{12} - \theta_2 - q_{21}) \\
&+ u_2u_6l1l2\cos(-q_{12} - \theta_2 - q_{21} - q_{22}) + 2u_3u_4(l1 + o_{n1})(o_{h2} + l2)\cos(-\theta_2) \\
&+ 2u_3u_5(l1 + o_{n1})l2\cos(-\theta_2 - q_{21}) + u_3u_6(l1 + o_{n1})l2\cos(-\theta_2 - q_{21} - q_{22}) \\
&+ 2u_4u_5(o_{h2} + l2)l2\cos(-q_{21}) + u_4u_6(o_{h2} + l2)l2\cos(-q_{21} - q_{22}) \\
&+ u_5u_6l2^2\cos(-q_{22}) \\
v_{l_2}^2 &= u_1^2(o_{h1} + l1)^2 + u_2^2l1^2 + u_3^2(l1 + o_{n1})^2 + u_4^2(o_{h2} + l2)^2 + u_5^2l2^2 + u_6^2(l2 + o_{r2} + c_{r2})^2 \\
&+ 2u_1u_2(o_{h1} + l1)l1\cos(-q_{11}) + 2u_1u_3(o_{h1} + l1)(l1 + o_{n1})\cos(-q_{11} - q_{12}) \\
&+ 2u_1u_4(o_{h1} + l1)(o_{h2} + l2)\cos(-q_{11} - q_{12} - \theta_2) + 2u_1u_5(o_{h1} + l1)l2\cos(-q_{11} - q_{12} - \theta - q_{21}) \\
&+ 2u_1u_6(o_{h1} + l1)(l2 + o_{r2} + c_{r2})\cos(-q_{11} - q_{12} - \theta - q_{21} - q_{22}) + 2u_2u_3l1(l1 + o_{n1})\cos(-q_{12}) \\
&+ 2u_2u_4l1(o_{h2} + l2)\cos(-q_{12} - \theta_2) + 2u_2u_5l1l2\cos(-q_{12} - \theta_2 - q_{21}) \\
&+ 2u_2u_6l1(l2 + o_{r2} + c_{r2})\cos(-q_{12} - \theta_2 - q_{21} - q_{22}) + 2u_3u_4(l1 + o_{n1})(o_{h2} + l2)\cos(-\theta_2) \\
&+ 2u_3u_5(l1 + o_{n1})l2\cos(-\theta_2 - q_{21}) + 2u_3u_6(l1 + o_{n1})(l2 + o_{r2} + c_{r2})\cos(-\theta_2 - q_{21} - q_{22}) \\
&+ 2u_4u_5(o_{h2} + l2)l2\cos(-q_{21}) + 2u_4u_6(o_{h2} + l2)(l2 + o_{r2} + c_{r2})\cos(-q_{21} - q_{22}) \\
&+ 2u_5u_6l2(l2 + o_{r2} + c_{r2})\cos(-q_{22})
\end{aligned}$$

In order to simplify notation, generalized inertia terms will be introduced at this point. The generalized inertia terms summarized below (for the 6 DOF TLA configuration only) will be used in the Lagrangian development for the dynamic equations of motion.

$$\begin{aligned}
J_{11} &= m_{h1}c_{h1}^2 + I_{h1} + m_{l1}\left(o_{h1} + \frac{l1}{2}\right)^2 + I_{l1} + (2m_{l1} + m_{r1} + m_{h2} + 3m_{r2} + m_{r2})(o_{h1} + l1)^2 \\
J_{22} &= m_{l1}\left(\frac{l1}{2}\right)^2 + I_{l1} + (m_{l1} + m_{n1} + m_{h2} + 3m_{r2} + m_{r2})l1^2 + I_{l1} \\
J_{33} &= m_{l1}\left(\frac{l1}{2}\right)^2 + I_{l1} + m_{l1}(l1 + o_{n1} + c_{n1})^2 + (m_{h2} + 3m_{r2} + m_{r2})(l1 + o_{n1})^2 \\
J_{44} &= m_{h2}c_{h2}^2 + I_{h2} + m_{l2}\left(o_{h2} + \frac{l2}{2}\right)^2 + I_{l2} + (2m_{l2} + m_{r2})(o_{h2} + l2)^2 \\
J_{55} &= m_{l2}\left(\frac{l2}{2}\right)^2 + I_{l2} + (m_{l2} + m_{r2})l2^2 \\
J_{66} &= m_{l2}\left(\frac{l2}{2}\right)^2 + I_{l2} + m_{l2}(l2 + o_{r2} + c_{r2})^2 + I_{l2} \\
J_{12} &= \frac{1}{2}[(3m_{l1} + 2m_{r1} + 2m_{h2} + 6m_{r2} + 2m_{r2})(o_{h1} + l1)l1] \\
J_{13} &= \frac{1}{2}[(m_{l1}l1 + 2m_{r1}(l1 + o_{n1} + c_{n1}) + (2m_{h2} + 6m_{r2} + 2m_{r2})(l1 + o_{n1}))(o_{h1} + l1)] \\
J_{14} &= \frac{1}{2}[(2m_{h2}c_{h2} + 2m_{r2}(o_{h2} + \frac{l2}{2}) + ((4m_{r2} + 2m_{r2})(o_{h2} + l2))(o_{h1} + l1)]
\end{aligned}$$

$$\begin{aligned}
J_{15} &= \frac{1}{2}[(3m_{i2} + 2m_{r2})(o_{h1} + l1)l2] \\
J_{16} &= \frac{1}{2}[(m_{i2}l2 + 2m_{r2}(l2 + o_{i2} + c_{i2}))(o_{h1} + l1)] \\
J_{23} &= \frac{1}{2}[(m_{h1}l1 + 2m_{i1}(l1 + o_{i1} + c_{i1}) + (2m_{h2} + 6m_{i2} + 2m_{r2})(l1 + o_{i1}))l1] \\
J_{24} &= \frac{1}{2}[(2m_{h2}c_{h2} + 2m_{i2}(o_{h2} + \frac{l2}{2}) + (4m_{i2} + 2m_{r2})(o_{h2} + l2))l1] \\
J_{25} &= \frac{1}{2}[(3m_{i2}l2(o_{h2} + l2) + 2m_{i2}l2)l1] \\
J_{26} &= \frac{1}{2}[(m_{i2}l2 + 2m_{r2}(l2 + o_{i2} + c_{i2}))l1] \\
J_{34} &= \frac{1}{2}[(2m_{h2}c_{h2} + 2m_{i2}(o_{h2} + \frac{l2}{2}) + (4m_{i2} + 2m_{r2})(o_{h2} + l2))(l1 + o_{i1})] \\
J_{35} &= \frac{1}{2}[(3m_{i2} + 2m_{r2})l2(l1 + o_{i1})] \\
J_{36} &= \frac{1}{2}[(m_{i2}l2 + 2m_{r2}(l2 + o_{i2} + c_{i2}))(l1 + o_{i1})] \\
J_{45} &= \frac{1}{2}[(3m_{i2} + 2m_{r2}))l2(o_{h2} + l2)] \\
J_{46} &= \frac{1}{2}[(m_{i2}l2 + 2m_{r2}(l2 + o_{i2} + c_{i2}))(o_{h2} + l2)] \\
J_{56} &= \frac{1}{2}[(m_{i2}l2 + 2m_{r2}(l2 + o_{i2} + c_{i2}))l2]
\end{aligned}$$

Using the generalized inertial terms presented above, the kinetic energy for the 6 DOF system may be represented as

$$\begin{aligned}
T = & \frac{J_{11}}{2}\dot{u}_1^2 + \frac{J_{22}}{2}\dot{u}_2^2 + \frac{J_{33}}{2}\dot{u}_3^2 + \frac{J_{44}}{2}\dot{u}_4^2 + \frac{J_{55}}{2}\dot{u}_5^2 + \frac{J_{66}}{2}\dot{u}_6^2 \\
& + J_{12}\dot{u}_1\dot{u}_2 \cos(u_1 - u_2) + J_{13}\dot{u}_1\dot{u}_3 \cos(u_1 - u_3) + J_{14}\dot{u}_1\dot{u}_4 \cos(u_1 - u_4) + J_{15}\dot{u}_1\dot{u}_5 \cos(u_1 - u_5) \\
& + J_{16}\dot{u}_1\dot{u}_6 \cos(u_1 - u_6) \\
& + J_{23}\dot{u}_2\dot{u}_3 \cos(u_2 - u_3) + J_{24}\dot{u}_2\dot{u}_4 \cos(u_2 - u_4) + J_{25}\dot{u}_2\dot{u}_5 \cos(u_2 - u_5) \\
& + J_{26}\dot{u}_2\dot{u}_6 \cos(u_2 - u_6) \\
& + J_{34}\dot{u}_3\dot{u}_4 \cos(u_3 - u_4) + J_{35}\dot{u}_3\dot{u}_5 \cos(u_3 - u_5) + J_{36}\dot{u}_3\dot{u}_6 \cos(u_3 - u_6) \\
& + J_{45}\dot{u}_4\dot{u}_5 \cos(u_4 - u_5) + J_{46}\dot{u}_4\dot{u}_6 \cos(u_4 - u_6) \\
& + J_{56}\dot{u}_5\dot{u}_6 \cos(u_5 - u_6)
\end{aligned}$$

The potential energy developed in the torsional springs is:

$$V = \frac{1}{2}k_1(u_2 - u_1)^2 + \frac{1}{2}k_1(u_3 - u_2)^2 + \frac{1}{2}k_2(u_5 - u_4)^2 + \frac{1}{2}k_2(u_6 - u_5)^2$$

Lagranges equation is of the form (B1) where  $Q'$  represents generalized forces due to the element torsional s damping.

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{u}_i} \right) - \frac{\partial T}{\partial u_i} = - \frac{\partial V}{\partial u_i} + Q_i \quad (\text{B1})$$

Carrying out this operation for the 6 DOF system, the following equations of motion were developed:

$$\begin{aligned}
& J_{11}\dot{u}_1 + J_{12}\dot{u}_2 \cos(-q_{11}) + J_{13}\dot{u}_3 \cos(-q_{11} - q_{12}) + J_{14}\dot{u}_4 \cos(-q_{11} - q_{12} - \theta_2) \\
& + J_{15}\dot{u}_5 \cos(-q_{11} - q_{12} - \theta_2 - q_{21}) + J_{16}\dot{u}_6 \cos(-q_{11} - q_{12} - \theta_2 - q_{21} - q_{22}) \\
& + J_{12}u_2^2 \sin(-q_{11}) + J_{13}u_3^2 \sin(-q_{11} - q_{12}) + J_{14}u_4^2 \sin(-q_{11} - q_{12} - \theta_2) \\
& + J_{15}u_5^2 \sin(-q_{11} - q_{12} - \theta_2 - q_{21}) + J_{16}u_6^2 \sin(-q_{11} - q_{12} - \theta_2 - q_{21} - q_{22}) \\
& = k_1(q_{11}) + b_1(u_2 - u_1) + T_1 \\
& J_{21}\dot{u}_1 \cos(-q_{11}) + J_{22}\dot{u}_2 + J_{23}\dot{u}_3 \cos(-q_{12}) + J_{24}\dot{u}_4 \cos(-q_{12} - \theta_2) \\
& + J_{25}\dot{u}_5 \cos(-q_{12} - \theta_2 - q_{21}) + J_{26}\dot{u}_6 \cos(-q_{12} - \theta_2 - q_{21} - q_{22}) \\
& - J_{21}u_1^2 \sin(-q_{11}) + J_{23}u_3^2 \sin(-q_{12}) + J_{24}u_4^2 \sin(-q_{12} - \theta_2) \\
& + J_{25}u_5^2 \sin(-q_{12} - \theta_2 - q_{21}) + J_{26}u_6^2 \sin(-q_{12} - \theta_2 - q_{21} - q_{22}) \\
& = -k_1(q_{11}) + k_1(q_{12}) - b_1(u_2 - u_1) + b_1(u_3 - u_2) \\
& J_{31}\dot{u}_1 \cos(-q_{11} - q_{12}) + J_{32}\dot{u}_2 \cos(-q_{12}) + J_{33}\dot{u}_3 + J_{34}\dot{u}_4 \cos(-\theta_2) \\
& + J_{35}\dot{u}_5 \cos(-\theta_2 - q_{21}) + J_{36}\dot{u}_6 \cos(-\theta_2 - q_{21} - q_{22}) \\
& - J_{31}u_1^2 \sin(-q_{11} - q_{12}) - J_{32}u_2^2 \sin(-q_{12}) + J_{34}u_4^2 \sin(-\theta_2) \\
& + J_{35}u_5^2 \sin(-\theta_2 - q_{21}) + J_{36}u_6^2 \sin(-\theta_2 - q_{21} - q_{22}) \\
& = -k_1(q_{12}) - b_1(u_3 - u_2) - T_2 \\
& J_{41}\dot{u}_1 \cos(-q_{11} - q_{12} - \theta_2) + J_{42}\dot{u}_2 \cos(-q_{12} - \theta_2) + J_{43}\dot{u}_3 \cos(-\theta_2) \\
& + J_{44}\dot{u}_4 + J_{45}\dot{u}_5 \cos(-q_{21}) + J_{46}\dot{u}_6 \cos(-q_{21} - q_{22}) \\
& - J_{41}u_1^2 \sin(-q_{11} - q_{12} - \theta_2) - J_{42}u_2^2 \sin(-q_{12} - \theta_2) - J_{43}u_3^2 \sin(-\theta_2) \\
& + J_{45}u_5^2 \sin(-q_{21}) + J_{46}u_6^2 \sin(-q_{21} - q_{22}) \\
& = +k_2(q_{21}) + b_2(u_5 - u_4) + T_2 \\
& J_{51}\dot{u}_1 \cos(-q_{11} - q_{12} - \theta_2 - q_{21}) + J_{52}\dot{u}_2 \cos(-q_{12} - \theta_2 - q_{21}) + J_{53}\dot{u}_3 \cos(-\theta_2 - q_{21}) \\
& + J_{54}\dot{u}_4 \cos(-q_{21}) + J_{55}\dot{u}_5 + J_{56}\dot{u}_6 \cos(-q_{22}) \\
& - J_{51}u_1^2 \sin(-q_{11} - q_{12} - \theta_2 - q_{21}) - J_{52}u_2^2 \sin(-q_{12} - \theta_2 - q_{21}) - J_{53}u_3^2 \sin(-\theta_2 - q_{21}) \\
& - J_{54}u_4^2 \sin(-q_{21}) + J_{56}u_6^2 \sin(-q_{22}) \\
& = -k_2(q_{21}) + k_2(q_{22}) - b_2(u_5 - u_4) + b_2(u_6 - u_5) \\
& J_{61}\dot{u}_1 \cos(-q_{11} - q_{12} - \theta_2 - q_{21} - q_{22}) + J_{62}\dot{u}_2 \cos(-q_{12} - \theta_2 - q_{21} - q_{22}) \\
& + J_{63}\dot{u}_3 \cos(-\theta_2 - q_{21} - q_{22}) + J_{64}\dot{u}_4 \cos(-q_{21} - q_{22}) + J_{65}\dot{u}_5 \cos(-q_{22}) + J_{66}\dot{u}_6 \\
& - J_{61}u_1^2 \sin(-q_{11} - q_{12} - \theta_2 - q_{21} - q_{22}) - J_{62}u_2^2 \sin(-q_{12} - \theta_2 - q_{21} - q_{22}) \\
& - J_{63}u_3^2 \sin(-\theta_2 - q_{21} - q_{22}) - J_{64}u_4^2 \sin(-q_{21} - q_{22}) - J_{65}u_5^2 \sin(-q_{22}) \\
& = -k_2(q_{22}) - b_2(u_6 - u_5)
\end{aligned}$$

## Appendix C: Model Poles and Zeros

### Full model poles

1.0000  
1.0000  
 $0.9219 + 0.3688i$   
 $0.9219 - 0.3688i$   
 $0.4687 + 0.8274i$   
 $0.4687 - 0.8274i$   
 $-0.1270 + 0.8837i$   
 $-0.1270 - 0.8837i$   
 $-0.5232 + 0.6723i$   
 $-0.5232 - 0.6723i$   
 $-0.7090 + 0.4354i$   
 $-0.7090 - 0.4354i$   
 $-0.7811 + 0.2632i$   
 $-0.7811 - 0.2632i$   
 $-0.8092 + 0.1393i$   
 $-0.8092 - 0.1393i$   
-0.8718  
-0.9416  
-0.7709  
-0.9665  
-0.9790  
-0.9660  
-0.9903  
-0.9929  
-0.9946  
-0.9958  
-0.9972  
-0.9965  
-0.9970  
-0.7127  
-0.6937  
-0.6846  
-0.6795  
-0.6765  
-0.6746  
-0.6734  
-0.6726  
-0.6718  
-0.6716  
-0.6721

### Poles reduced model

$0.9219 + 0.3688i$   
 $0.9219 - 0.3688i$   
 $0.4687 + 0.8274i$   
 $0.4687 - 0.8274i$

$-0.1287 + 0.8806i$   
 $-0.1287 - 0.8806i$   
 $-0.5779 + 0.6116i$   
 $-0.5779 - 0.6116i$   
 $1.0000 + 0.0000i$   
 $1.0000 - 0.0000i$

### Zeros of Reduced System

$n = 1$

**ans =**  
 $-1.0000$   
 $-0.5362 + 0.6474i$   
 $-0.5362 - 0.6474i$   
 $0.0038 + 0.9042i$   
 $0.0038 - 0.9042i$   
 $0.7576 + 0.6175i$   
 $0.7576 - 0.6175i$   
 $0.9960 + 0.0807i$   
 $0.9960 - 0.0807i$   
 $1.0000$

$n = 2$

**ans =**  
 $-1.0000$   
 $-0.4373 + 0.7841i$   
 $-0.4373 - 0.7841i$   
 $0.2227 + 0.9273i$   
 $0.2227 - 0.9273i$   
 $0.9065 + 0.4420i$   
 $0.9065 - 0.4420i$   
 $1.1025$   
 $1.0000 + 0.0000i$   
 $1.0000 - 0.0000i$

$n = 3$

**ans =**  
 $-1.0000$   
 $-0.4581 + 0.6520i$   
 $-0.4581 - 0.6520i$   
 $0.1342 + 0.8911i$   
 $0.1342 - 0.8911i$   
 $0.8608 + 0.4546i$   
 $0.8608 - 0.4546i$   
 $0.9452$   
 $1.0000 + 0.0000i$   
 $1.0000 - 0.0000i$

$n = 4$

ans =  
-1.0000  
0.0223 + 0.8748i  
0.0223 - 0.8748i  
-0.5206 + 0.3255i  
-0.5206 - 0.3255i  
0.8174 + 0.4686i  
0.8174 - 0.4686i  
1.0000 + 0.0000i  
1.0000 - 0.0000i  
0.8762

n = 5

ans =  
-1.4463  
-1.0000  
-0.0766 + 0.8838i  
-0.0766 - 0.8838i  
0.7867 + 0.5096i  
0.7867 - 0.5096i  
1.0000 + 0.0000i  
1.0000 - 0.0000i  
0.8986  
-0.1107

n = 6

ans =  
-3.3932  
-1.0000  
-0.1547 + 0.8677i  
-0.1547 - 0.8677i  
0.7769 + 0.5739i  
0.7769 - 0.5739i  
1.0000 + 0.0000i  
1.0000 - 0.0000i  
0.9736  
-0.0700

n = 7

ans =  
37.0422  
-1.0000  
-0.2114 + 0.8076i  
-0.2114 - 0.8076i  
0.7677 + 0.6524i  
0.7677 - 0.6524i  
1.0579  
1.0000 + 0.0000i  
1.0000 - 0.0000i  
-0.1427

**n = 8**

ans =  
3.2727  
-1.0000  
-0.2393 + 0.7083i  
-0.2393 - 0.7083i  
0.7158 + 0.7411i  
0.7158 - 0.7411i  
1.1348  
1.0000 + 0.0000i  
1.0000 - 0.0000i  
-0.2552

**n = 9**

ans =  
2.0600  
-1.0000  
0.5948 + 0.7962i  
0.5948 - 0.7962i  
-0.2336 + 0.5534i  
-0.2336 - 0.5534i  
1.1768  
1.0000 + 0.0000i  
1.0000 - 0.0000i  
-0.4071

**n = 10**

ans =  
1.8731  
-1.0000  
0.4384 + 0.8173i  
0.4384 - 0.8173i  
1.1424  
1.0000 + 0.0000i  
1.0000 - 0.0000i  
-0.7084  
-0.1438 + 0.2757i  
-0.1438 - 0.2757i

**n = 11**

ans =  
1.9851  
0.2871 + 0.8670i  
0.2871 - 0.8670i  
-1.1711  
1.0562  
1.0000 + 0.0000i

1.0000 - 0.0000i  
-1.0000  
0.3401  
-0.3808

n = 12

ans =  
-1.8434  
2.2340  
-1.0000  
0.1725 + 0.9158i  
0.1725 - 0.9158i  
1.0000 + 0.0000i  
1.0000 - 0.0000i  
0.9506  
0.6072  
-0.5093

n = 13

ans =  
-3.2719  
2.5785  
-1.0000  
0.0632 + 0.9284i  
0.0632 - 0.9284i  
1.0000 + 0.0000i  
1.0000 - 0.0000i  
0.8129 + 0.1045i  
0.8129 - 0.1045i  
-0.6002

n = 14

ans =  
-8.5635  
2.9285  
-0.0480 + 0.8897i  
-0.0480 - 0.8897i  
-1.0000  
0.8292 + 0.1694i  
0.8292 - 0.1694i  
1.0000 + 0.0000i  
1.0000 - 0.0000i  
-0.6783

n = 15

ans =  
-80.2913  
2.8908

$-0.1512 + 0.7925i$   
 $-0.1512 - 0.7925i$   
 $-1.0000$   
 $0.8360 + 0.1815i$   
 $0.8360 - 0.1815i$   
 $1.0000 + 0.0000i$   
 $1.0000 - 0.0000i$   
 $-0.7538$

**n =16**

**ans =**  
 $-37.8924$   
 $2.2999$   
 $-1.0000$   
 $-0.8513$   
 $-0.2462 + 0.6132i$   
 $-0.2462 - 0.6132i$   
 $1.0000 + 0.0000i$   
 $1.0000 - 0.0000i$   
 $0.8353 + 0.1546i$   
 $0.8353 - 0.1546i$

**n =17**

**ans =**  
 $-7.4878$   
 $1.7132$   
 $1.0000 + 0.0000i$   
 $1.0000 - 0.0000i$   
 $-1.1745$   
 $0.9357$   
 $-1.0000$   
 $0.7148$   
 $-0.2925 + 0.1113i$   
 $-0.2925 - 0.1113i$

**n =18**

**ans =**  
 $-2.0521 + 1.3866i$   
 $-2.0521 - 1.3866i$   
 $-1.0000$   
 $-0.7280$   
 $1.1947 + 0.2069i$   
 $1.1947 - 0.2069i$   
 $1.0000 + 0.0000i$   
 $1.0000 - 0.0000i$   
 $0.4193 + 0.2455i$   
 $0.4193 - 0.2455i$

**n =19**

ans =  
-1.2694 + 1.8730i  
-1.2694 - 1.8730i  
-1.0000  
-0.7990  
1.0954 + 0.3460i  
1.0954 - 0.3460i  
1.0000 + 0.0000i  
1.0000 - 0.0000i  
0.4804 + 0.4289i  
0.4804 - 0.4289i

n =20

ans =  
-0.8774 + 1.9440i  
-0.8774 - 1.9440i  
-1.0000  
-0.8335  
0.5182 + 0.5407i  
0.5182 - 0.5407i  
1.0213 + 0.3833i  
1.0213 - 0.3833i  
1.0000 + 0.0000i  
1.0000 - 0.0000i

n =21

ans =  
0.0045 + 0.9045i  
0.0045 - 0.9045i  
-0.5351 + 0.6492i  
-0.5351 - 0.6492i  
0.7582 + 0.6177i  
0.7582 - 0.6177i  
0.9964 + 0.0807i  
0.9964 - 0.0807i  
-1.0079  
-1.0000

n =22

ans =  
0.2056 + 0.9063i  
0.2056 - 0.9063i  
-0.4304 + 0.7461i  
-0.4304 - 0.7461i  
0.8879 + 0.4379i  
0.8879 - 0.4379i  
-1.0000  
1.0000 + 0.0000i

1.0000 - 0.0000i  
 -0.8955

n =

23

ans =  
 -1.0941  
 -1.0000  
 -0.4623 + 0.6892i  
 -0.4623 - 0.6892i  
 0.1478 + 0.9052i  
 0.1478 - 0.9052i  
 0.8741 + 0.4614i  
 0.8741 - 0.4614i  
 1.0000 + 0.0000i  
 1.0000 - 0.0000i

n = 24

ans =  
 -1.4097  
 -1.0000  
 0.0695 + 0.8988i  
 0.0695 - 0.8988i  
 -0.5004 + 0.5794i  
 -0.5004 - 0.5794i  
 0.8553 + 0.4910i  
 0.8553 - 0.4910i  
 1.0000 + 0.0000i  
 1.0000 - 0.0000i

n = 25

ans =  
 -2.0625  
 0.8289 + 0.5291i  
 0.8289 - 0.5291i  
 1.0000 + 0.0000i  
 1.0000 - 0.0000i  
 -0.0312 + 0.8854i  
 -0.0312 - 0.8854i  
 -1.0000  
 -0.5286 + 0.3561i  
 -0.5286 - 0.3561i

n = 26

ans =  
 -3.8851

-1.0000  
 -0.8903  
 -0.1420 + 0.8628i  
 -0.1420 - 0.8628i  
 0.7909 + 0.5785i  
 0.7909 - 0.5785i  
 1.0000 + 0.0000i  
 1.0000 - 0.0000i  
 -0.1885

n = 27

ans =  
 -18.4715  
 -1.1711  
 -1.0000  
 -0.2419 + 0.8238i  
 -0.2419 - 0.8238i  
 0.7355 + 0.6412i  
 0.7355 - 0.6412i  
 1.0000 + 0.0000i  
 1.0000 - 0.0000i  
 0.0672

n =

28

ans =  
 8.5008  
 -1.3747  
 -1.0000  
 -0.3257 + 0.7643i  
 -0.3257 - 0.7643i  
 0.6558 + 0.7149i  
 0.6558 - 0.7149i  
 1.0000 + 0.0000i  
 1.0000 - 0.0000i  
 0.2274

n = 29

ans =  
 3.8912  
 -1.5386  
 -1.0000  
 -0.4022 + 0.6816i  
 -0.4022 - 0.6816i  
 0.5496 + 0.7879i  
 0.5496 - 0.7879i  
 1.0000 + 0.0000i  
 1.0000 - 0.0000i

0.3405

n = 30

```
ans =
2.7544
-1.6497
0.4252 + 0.8460i
0.4252 - 0.8460i
1.0000 + 0.0000i
1.0000 - 0.0000i
-1.0000
-0.4851 + 0.5622i
-0.4851 - 0.5622i
0.4215
```

n = 31

```
ans =
2.2717
-1.6581
0.2945 + 0.8843i
0.2945 - 0.8843i
1.0000 + 0.0000i
1.0000 - 0.0000i
-1.0000
-0.5997 + 0.3426i
-0.5997 - 0.3426i
0.4801
```

n = 32

```
ans =
2.0147
-1.3683 + 0.2834i
-1.3683 - 0.2834i
0.1630 + 0.9036i
0.1630 - 0.9036i
1.0000 + 0.0000i
1.0000 - 0.0000i
-1.0000
0.5240
-0.3205
```

n = 33

```
ans =
-1.5365 + 0.6558i
-1.5365 - 0.6558i
1.8581
-1.0000
0.0306 + 0.9017i
```

0.0306 - 0.9017i  
1.0000 + 0.0000i  
1.0000 - 0.0000i  
0.5579  
-0.1590

n =34

ans =  
-1.6747 + 0.9069i  
-1.6747 - 0.9069i  
1.7544  
-1.0000  
-0.1051 + 0.8707i  
-0.1051 - 0.8707i  
1.0000 + 0.0000i  
1.0000 - 0.0000i  
0.5847  
-0.0745

n =35

ans =  
-1.8549 + 1.0999i  
-1.8549 - 1.0999i  
1.6822  
-1.0000  
-0.2445 + 0.7989i  
-0.2445 - 0.7989i  
1.0000 + 0.0000i  
1.0000 - 0.0000i  
0.6061  
-0.0254

n =36

ans =  
-2.2028 + 1.1223i  
-2.2028 - 1.1223i  
1.6304  
-1.0000  
-0.3923 + 0.6688i  
-0.3923 - 0.6688i  
1.0000 + 0.0000i  
1.0000 - 0.0000i  
0.6232  
0.0016

n =37

ans =  
-4.2350

-1.9434  
 1.5923  
 -1.0000  
 -0.6074 + 0.3680i  
 -0.6074 - 0.3680i  
 1.0000 + 0.0000i  
 1.0000 - 0.0000i  
 0.6369  
 0.0104

n = 38

ans =  
 -15.9546  
 -1.2629 + 0.7633i  
 -1.2629 - 0.7633i  
 -1.0000  
 1.5624  
 1.0000 + 0.0000i  
 1.0000 - 0.0000i  
 0.6483  
 -0.1235  
 -0.0235

n = 39

ans =  
 15.3484  
 -1.2533 + 1.0041i  
 -1.2533 - 1.0041i  
 -1.0000  
 1.5334  
 1.0000  
 1.0000  
 0.6599  
 0.0319 + 0.1043i  
 0.0319 - 0.1043i

n = 40

ans =  
 5.1616  
 -1.2461 + 1.1384i  
 -1.2461 - 1.1384i  
 -1.0000  
 1.4893  
 1.0000  
 1.0000  
 0.6782  
 0.1139 + 0.0910i  
 0.1139 - 0.0910i

**n =41**

ans =  
-5.6194  
1.6385  
-1.3450 + 0.9588i  
-1.3450 - 0.9588i  
-0.2103 + 0.8379i  
-0.2103 - 0.8379i  
-1.0000  
0.6248  
-0.0871 + 0.1539i  
-0.0871 - 0.1539i

**n =42**

ans =  
-5.6194  
1.6385  
-1.3450 + 0.9588i  
-1.3450 - 0.9588i  
-0.2103 + 0.8379i  
-0.2103 - 0.8379i  
-1.0000  
0.6248  
-0.0871 + 0.1539i  
-0.0871 - 0.1539i

### Appendix D: Bode/Root Locus Plots

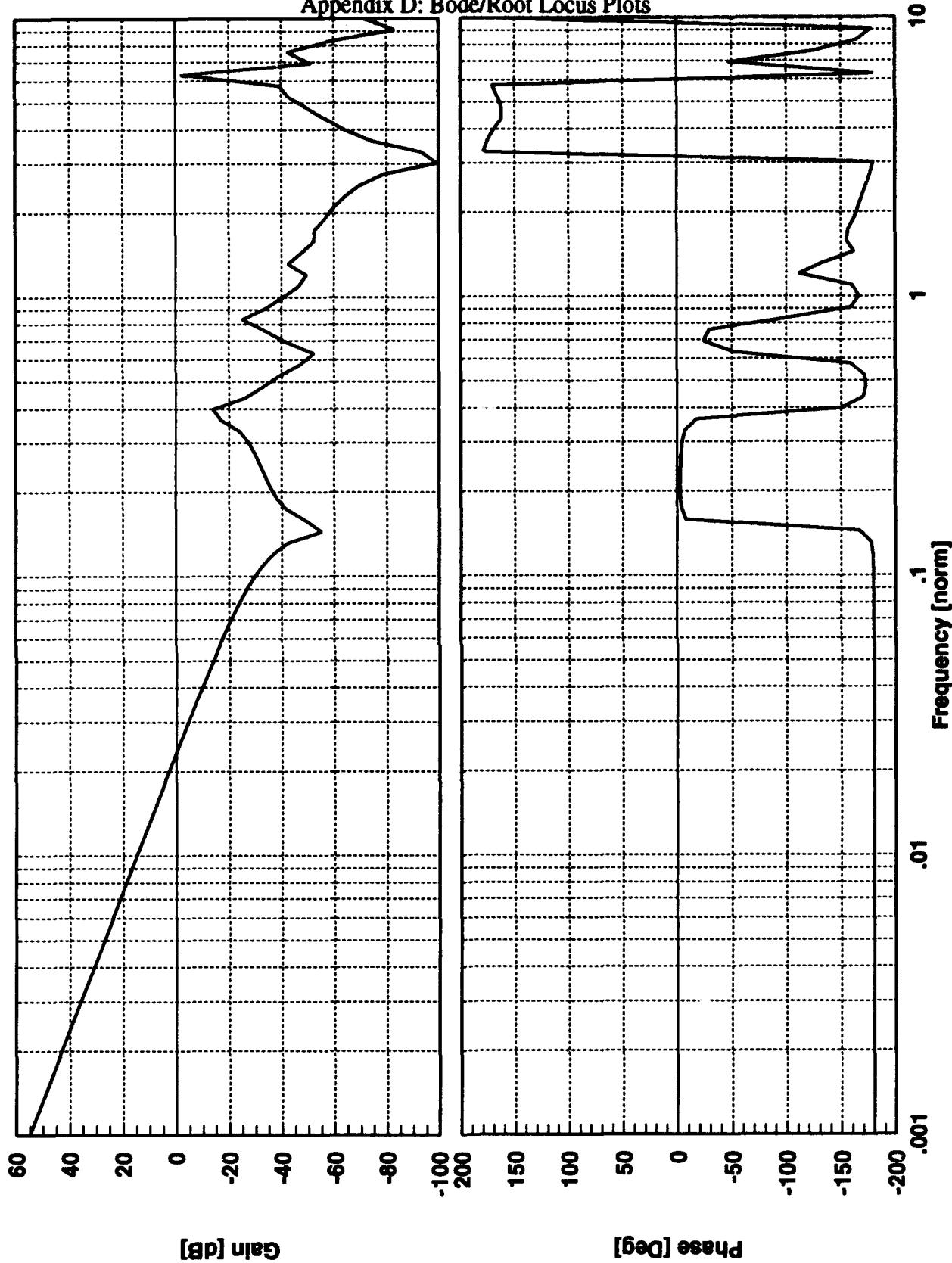


Figure D.1 Open Loop On-Design Bode Plot

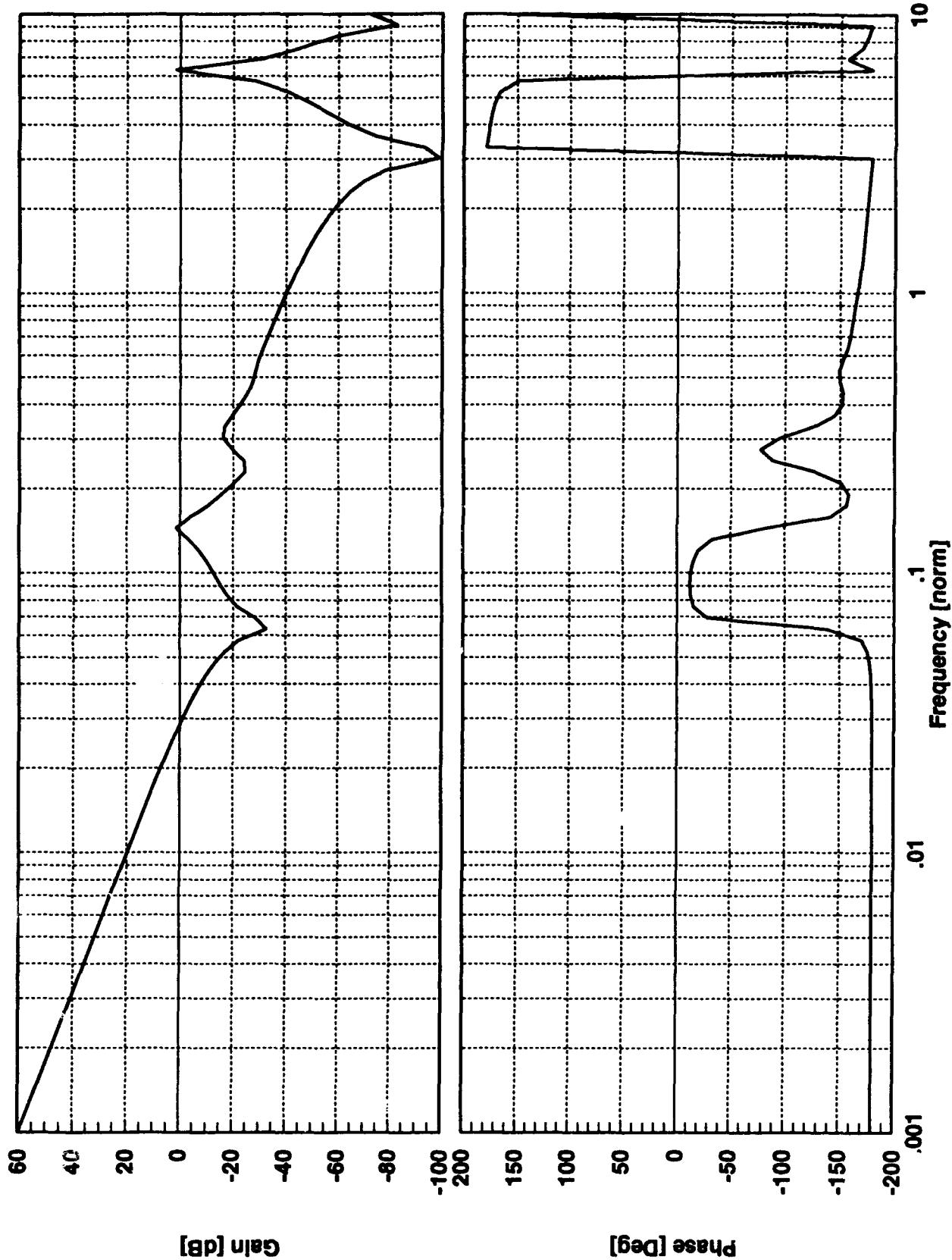


Figure D.2 Open Loop Off-Design Flexible Link Bode Plot

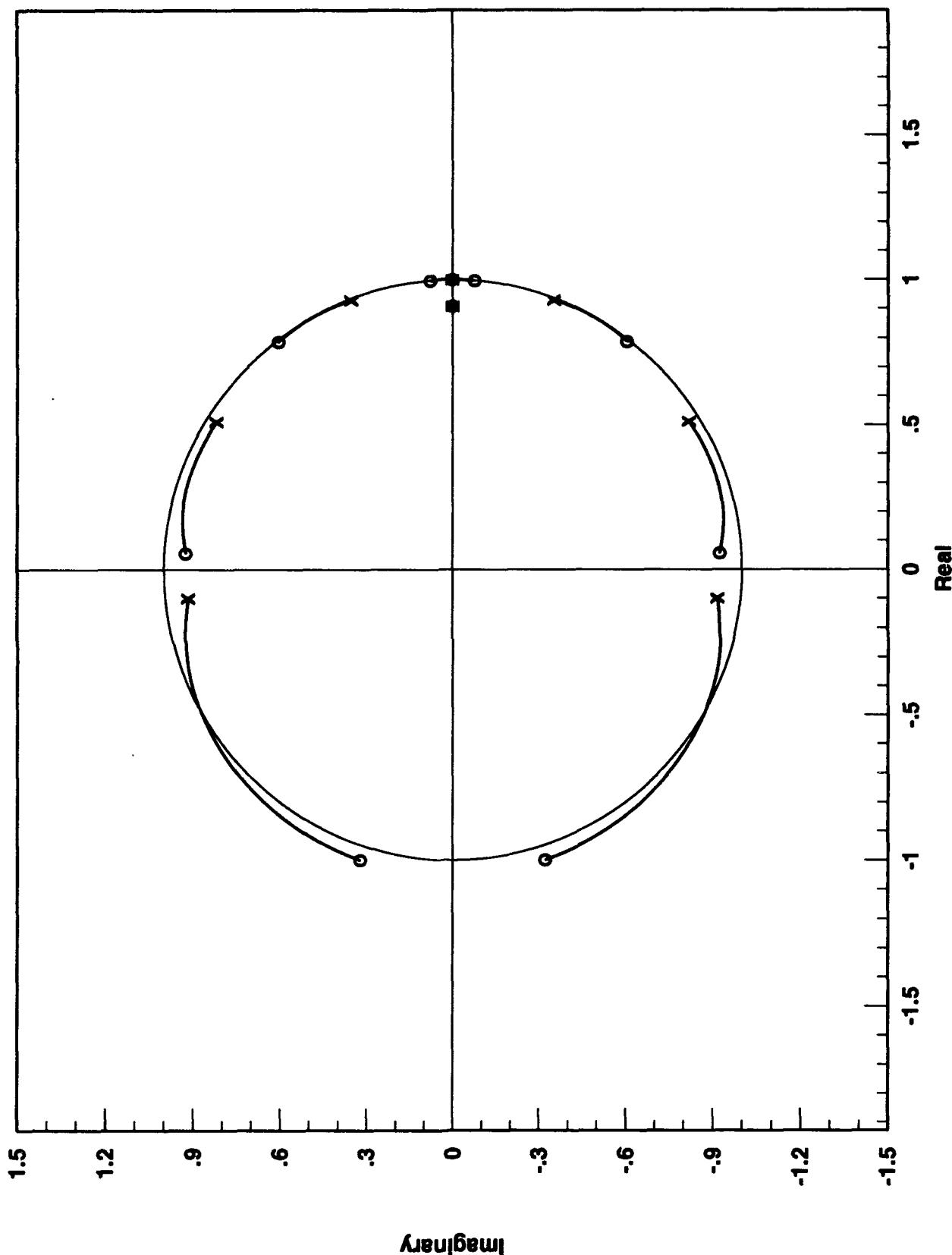


Figure D.3 Open Loop On-Design Root Locus

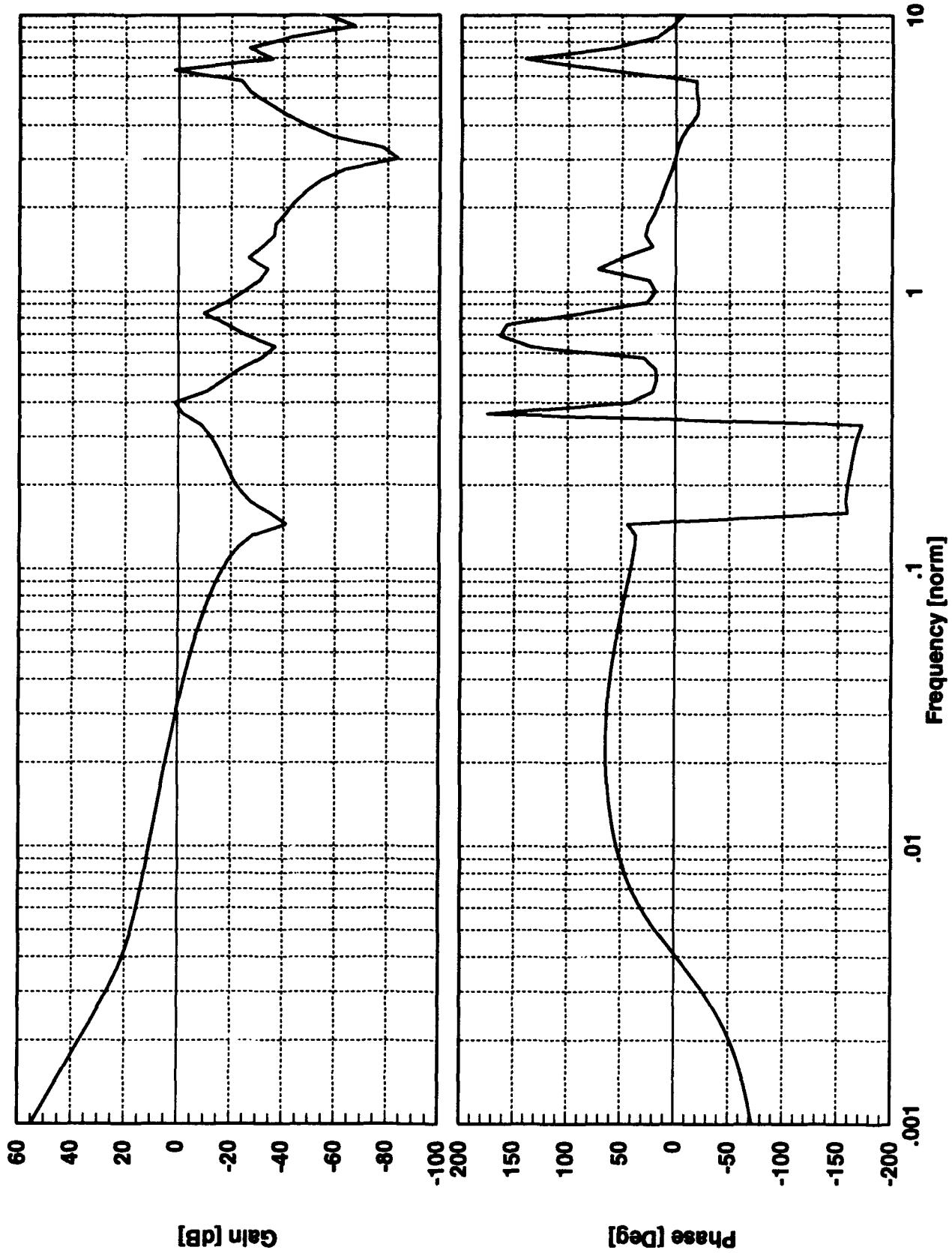


Figure D.4 Closed Loop PID On-Design Bode Plot

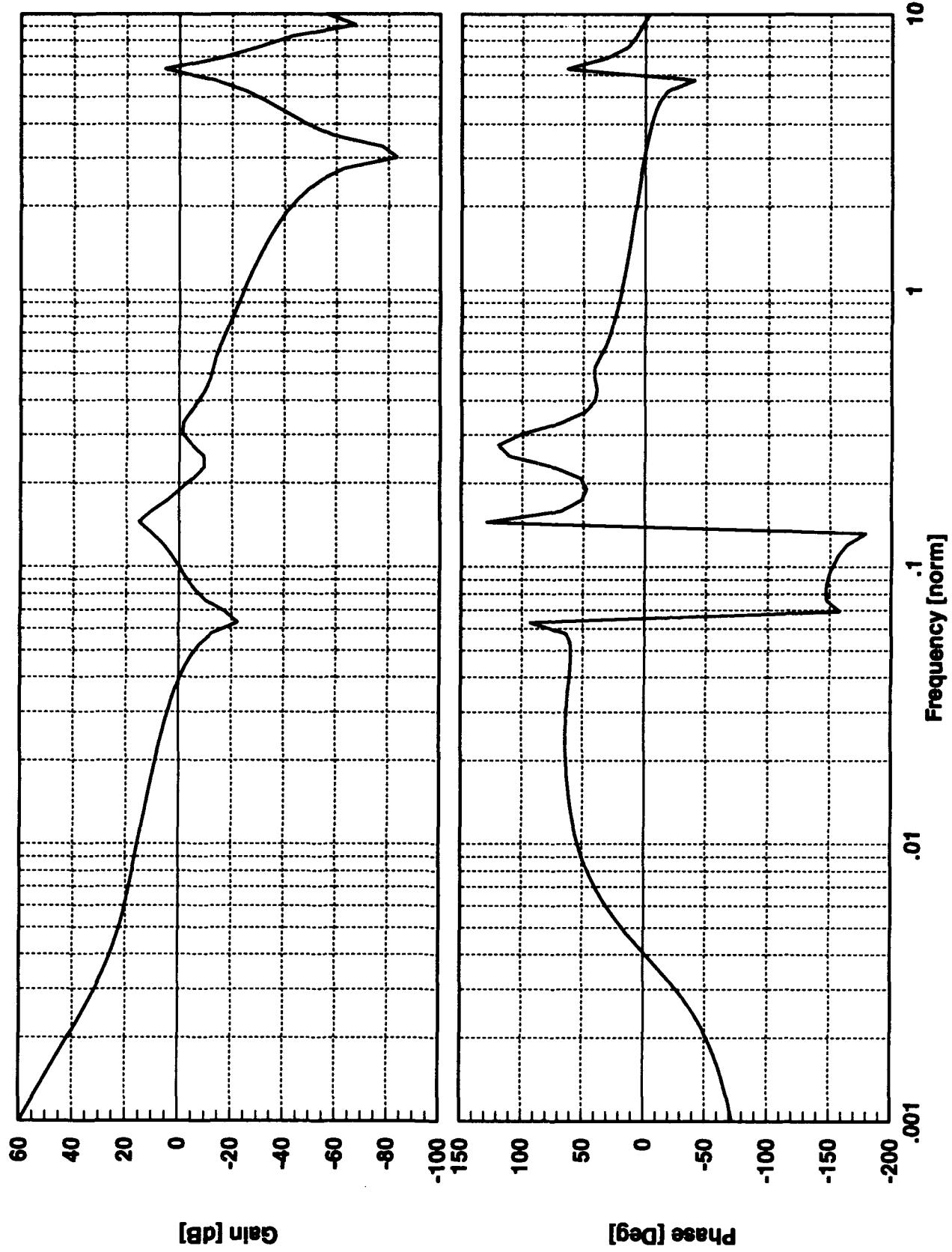


Figure D.5 Closed Loop PID Off-Design Flexible Link Bode Plot

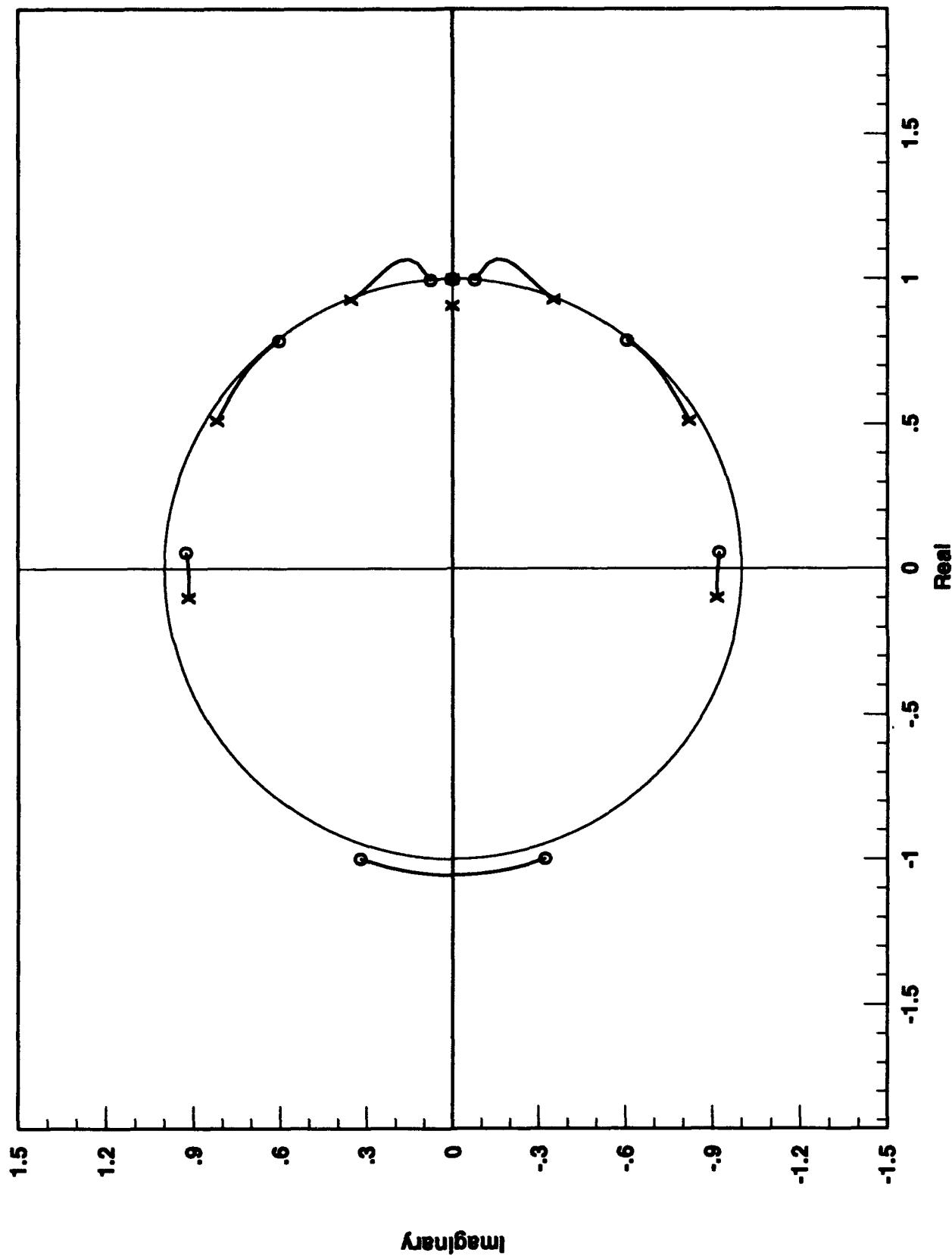


Figure D.6 Closed Loop PID On-Design Root Locus